





UNIVERSITY OF  
ILLINOIS LIBRARY  
AT URBANA-CHAMPAIGN



## CENTRAL CIRCULATION BOOKSTACKS

The person charging this material is responsible for its renewal or its return to the library from which it was borrowed on or before the **Latest Date** stamped below. **You may be charged a minimum fee of \$75.00 for each lost book.**

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

TO RENEW CALL TELEPHONE CENTER, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

FEB 21 1976

AUG 7 2000

DEC 07 2010

When renewing by phone, write new due date below  
previous due date.

L162



IL 6r  
No. 982

UIUCDCS-R-79-982

Copy 2

UILU-ENG 79 1730

Learning Without Negative Examples via Variable-Valued  
Logic Characterizations: The Uniclass Inductive Program AQ7UNI

by

ROBERT STEPP

July 1979



DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS



Report No. UIUCDCS-R-79-982

Learning Without Negative Examples via  
Variable-Valued Logic Characterizations:  
The Uniclass Inductive Program AQ7UNI

by

Robert Stepp

July 1979

Department of Computer Science  
University of Illinois at Urbana-Champaign  
Urbana, Illinois 61801

This work was supported in part by the National Science  
Foundation under Grant NSF MCS 79-06614.





## ABSTRACT

This paper describes the underlying theory, internal logic, and evaluation of an inductive program AQ7UNI which accepts a set of symbolic descriptions (events) of arbitrary objects and produces a general description (characterization) of the set. The events are attribute-value lists and the resulting characterizations are expressed in a simple yet powerful formal language VL<sub>1</sub> (Variable-valued Logic system 1 [Michalski 74, 75]), which is a form of monadic predicate calculus.

AQ7UNI belongs to a family of inductive programs developed at the University of Illinois, Department of Computer Science (see [Michalski 77a, 77b] for a summary) which employ quasi-extremal optimality techniques. The degree of generalization (as defined in [Michalski 79]) and the search space used in this method can be controlled by the user, through a variety of operational parameters entered with the problem data.

Several artificially-constructed problems from recent papers on variable-valued logic ([Michalski 75, 78] and [Larson 77]) are used to illustrate the program's capabilities.

The author acknowledges the support and encouragement provided by R. S. Michalski, Associate Professor of Computer Science, University of Illinois, who was the source of the fundamental algorithms on which this work is based.



Digitized by the Internet Archive  
in 2013

<http://archive.org/details/learningwithoutn982step>

## C O N T E N T S

Introduction	1
The Uniclass Algorithm	6
Sample Characterizations	
"TRAINS"	16
"BOTTLES"	20
"FACES"	25
"ANIMALS"	28
Comparison with Other Methods	43
Summary	55



## Introduction

This report describes a technique for constructing characteristic descriptions of a set of instances of a class of objects (events). It is assumed that negative examples, i.e. examples of objects not belonging to the class, are not available. Thus, the concept formation problems being considered are somewhat more difficult than those in which negative examples are present, particularly if they are carefully selected "near misses" (e.g. [Winston 70]). The problem of learning from only positive examples is called a uniclass generalization problem.

The characteristic descriptions or characterizations will be in the form of logical rules defined in the variable-valued logic system VL<sub>1</sub> [Michalski 74]. The characterizations can be very explicit, retaining all features of all events, or they can be very general, referring to just a few key features. This difference is captured mathematically by an attribute of a characterization called the degree of generalization, which will be precisely defined shortly. At the lowest degree of generalization, a characterization will retain and separate the features of every event in the group, and is just a list of the attributes of every event. At a medium degree of generalization, the characterization will make a statement of general attributes about several subgroups of similar events. Finally, at the highest degree of generalization, the characterization gives just one statement which gives general attributes for all events lumped together.

There are many inductive problems in which specific instances of a class of objects are given and the task is to determine the common properties of the objects in the class. Sometimes a taxonomy of the feature space is desired, based on key features, such as those features present in characterizations of medium degrees of generalization. The subgroupings of the events used in the characterization form clusters of similar events, each cluster differing from the others. Such ad hoc clustering may reveal a fundamental property of the system supplying the events, or at least aid feature selection by pointing out variables of special interest.

The work described in this paper utilizes an event description language called  $VL_1$ , which was developed by R. S. Michalski. Actually only a subset of  $VL_1$  is necessary for this work. Using this subset, we assume that all features are representable by non-negative integers and that for each feature there is a positive integer  $d$  which gives the number of levels in its domain, i.e. the feature value will be in the closed interval  $[0, d-1]$ . Unless specially named, features are referenced by the variables  $X_1, X_2, X_3$ , and so on.

In system  $VL_1$ , events are described by a logical expression called a term which is the product (logical conjunction) of selectors. Each selector is a logical unit of the form  $[X_i = \text{set of values}]$  and is true when the value of the variable  $X_i$  is an element in the set of values. The set of values is the reference set of variable  $X_i$ .

In  $VL_1$ , a selector may utilize other relational operators  $\neg$ ,  $=$ ,  $<$ ,  $>$  as well as  $=$ , however the equality relationship is the only one included in the subset of  $VL_1$  which is necessary for the task at hand. An event with  $n$  features represented by the variables  $X_1, X_2, \dots, X_n$  is represented in system  $VL_1$  as the product of  $n$  selectors, one selector for each variable. The reference sets all contain but a single value, such that the product of selectors is true only for the feature values of the event it represents. To illustrate this, suppose there are three features and a specific event is given by the triple  $(3,4,2)$ . The  $VL_1$  expression  $[X_1=3][X_2=4][X_3=2]$  is satisfied only by the event  $(3,4,2)$  and uniquely represents this event in the  $VL_1$  language. The  $VL_1$  expression  $[X_1=3][X_3=2]$  is also satisfied by event  $(3,4,2)$  and perhaps many other events as well because no selector for  $X_2$  occurs. When a selector is omitted, the missing variables are free to assume any value in their domains. If the number of levels in the domain of  $X_2$  is 6, then the preceding  $VL_1$  expression is equivalent to

$$[X_1=3][X_2=0,1,2,3,4,5][X_3=2].$$

Algorithms and their computer implementations exist for synthesizing variable-valued logic ( $VL_1$ ) expressions to discriminate between two or more classes of events [Michalski 73,74] [Larson & Michalski 75]. A different problem exists when a  $VL_1$  characterization of a single class of events is sought since no negative examples are present. Such a characterization may also be called the uniclass cover of the class of events.



When producing VL<sub>1</sub> expressions which will discriminate between two or more classes, the expression for any given class must describe a subset of the event space which includes or covers the events in the learning sample for that class while excluding alien events belonging to any other class. Better covers are those which are simpler or more meaningful as reflected in the number of terms in the VL<sub>1</sub> expression, the number of selectors they contain, as well as other measures of desirability which the programmer may select. An important point is that even the worst cover must still include all events in the learning sample while excluding all events in any other class. If only a single class of events is present, the test for the inclusion of all learning events with simultaneous exclusion of alien events cannot be applied. A uniclass problem may be converted into a two class problem by inventing a second class of events defined to be all those events in the event space not belonging to the first class of events, but such an approach, which produces an exact uniclass cover, is of the lowest degree of generalization and does not generally yield a description of the class of events any simpler or more meaningful than was given in the original event data. Simpler and more meaningful are terms which do not have precise meanings at present, but one can develop precise criteria



of optimality of solutions which can approximate the simplicity or meaningfulness.

In the multi-class approach, one wants to generate the simplest discrimination rules possible which implies that  $VL_1$  rules should cover as large a subspace as possible, but of course still not cover any alien events. That approach to the uniclass problem yields two dilemmas. First, in the "exact cover" case, all points in the event space which are not sample event points are alien events and initial covers (that form the characterization of the lowest degree of generalization) cannot be expanded and generalized since doing so would include an alien point. As the result, no generalization can be made and only a simplification of the original data is possible, i.e. the number of terms in the characterization may not be as great as the number of events, but the complexity of the characterization expression remains maximal.

Second, in the "approximate cover" case, no alien points in the event space are created and the expansion of the  $VL_1$  rules to include adjacent points in the event space is allowed. The attempt to make the terms as general as possible to simplify them takes one to the extreme, when entire event space is covered. The result is too much generalization and the  $VL_1$  characterization no longer shows any individual features of the learning events. If an advantageous characterization of a class of events exists, its degree of generalization must lie somewhere between

the two extremes mentioned above. In the solution of this dilemma, the degree of generalization is controlled by introducing the concepts of density threshold and selector threshold to determine where the middle ground on term generalization is to be taken.

### The Uniclass Algorithm

The basic uniclass algorithm was developed by R. S. Michalski and was partially implemented by a student at the University of Illinois, H. Yuen. The following discussion will explain the algorithm as it exists in a modified and extended form which is the basis for the implementation of the inductive program AQ7UNI, version 2.

In the VL<sub>1</sub> system [Michalski 74], a cover for a class of events is a logical formula which is the disjunction of logical expressions called terms. It has already been shown, by examples, how a term is the product of selectors, and how a term may be satisfied by one or more events. A complex is a subset of the set of all points in the event space. For every term there is an associated complex composed of all those points in the event space at which the term is satisfied. Some complexes cannot be represented by a single term. Such complexes will be purposely avoided by requiring that any complex be exactly described by some term, and with this constraint

terms and complexes become equivalent, one making a logical statement, the other a set-theoretical statement, about the same situation. Throughout the remainder of this paper the words "term" and "complex" will be used interchangeably, each connoting the hidden properties of the other.

A cover is a set of complexes (a list of terms) such that every event is in the union of the complexes. If the intersection of any two distinct complexes is non-empty, the complexes are intersecting, otherwise they are disjoint.

The variables in system  $VL_1$  may be of nominal or interval scale. Nominal scale variables may be simple, called FACTOR type variables, or generalization tree structured, called STRUCTURED type variables. Interval scale variables are called INTERVAL type variables. A syntactic limitation is built into the  $VL_1$  selector reference set notation. FACTOR variable reference sets may be any powerset of the domain of the variable. STRUCTURED variable reference sets must be a single leaf or node in the generalization tree. INTERVAL variable reference sets must be a single interval subset of the domain of the variable. Because of term/complex equivalency, these syntactic restrictions also further restrict the subsets of events which are legal complexes.

Some definitions are needed to assist with a formal presentation of the uniclass algorithm.

variable  $X_i$

$X_i$   $1 \leq i \leq n$ , is an integer-valued variable representing some feature of an event.

event  $e$

$e \equiv \{x_1, x_2, \dots, x_n\}$  is a sequence of  $n$  values for the  $n$  corresponding variables

event set  $E$

$E \equiv \{e_1, e_2, \dots, e_m\}$  is a set of  $m$  events

selector reference set  $r_i$

$r_i \equiv \{v_{i1}, v_{i2}, \dots, v_{is_i}\}$   $1 \leq i \leq n$ , is a set of  $s_i$  values in the domain of variable  $X_i$

selector  $[S_i]$

$[S_i] \equiv [X_i = r_i]$  is a logical expression in the  $VL_1$  system which is true only when the value of variable  $X_i$  is an element of the set  $r_i$

complex  $C(E)$

$C(E) \equiv [S_{k_1}][S_{k_2}] \dots [S_{k_j}]$   $1 \leq j \leq n$   
is a conjunction of selectors which is true for all events in set  $E$  and false for the maximum number of events not in set  $E$ .  $C(E)$  is both a term in a  $VL_1$  logical expression and the corresponding subspace of the event space.

density of complex  $(D(C(E)))$

$D(C(E)) \equiv \frac{\#E}{\#C}$  is the ratio of the number of events in set  $E$  to the number of points in the event subspace  $C(E)$ .

degree of generalization  $AG(E)$

$AG(E) \equiv -\log_2 D(C(E))$  (introduced in [Michalski 79]) is the average number of bits of information needed to locate a particular event from  $E$  within a unit event subspace of size  $\#C/\#E$ , with  $\#E$  and

#C as defined in the previous definition of density. When event set E is described by complex C, each event in E is being described by an enclosing subspace containing an average of #C/#E points. This gives "generality" or "uncertainty" to the location of the event (it is one of the #C/#E points, but which one is it?). The degree of generalization is the average amount of information disregarded in determining the location of the event when it is described by C(E) rather than E.

### rank R(e,e')

R is a measure of the dissimilarity of the events e and e'. Recall that both e and e' are sequences of n values for the n variables. Let  $d(x,x')$  be 0 if  $x=x'$  and 1 otherwise. Then

$$R(e,e') = \sum_{i=1}^n d(x_i, x'_i) \quad \text{where } x_i \quad 1 \leq i \leq n \text{ is the}$$

sequence of values for e and  $x'_i \quad 1 \leq i \leq n$  is the corresponding sequence for e'. Rank R is a special case of a more complete dissimilarity measure for VL1 events found in [Michalski & Larson 78] which applies to nominal, interval, and structured variables. The dissimilarity measure used in this report corresponds to the general treatment the authors cited give to nominal scale variables.

The general uniclass algorithm is diagrammed in figure 1. The symbolic notation used is defined below.

$E_L$  - the set of events

$E_R$  - the set of events remaining to be covered ( $E_R \subseteq E_L$ )

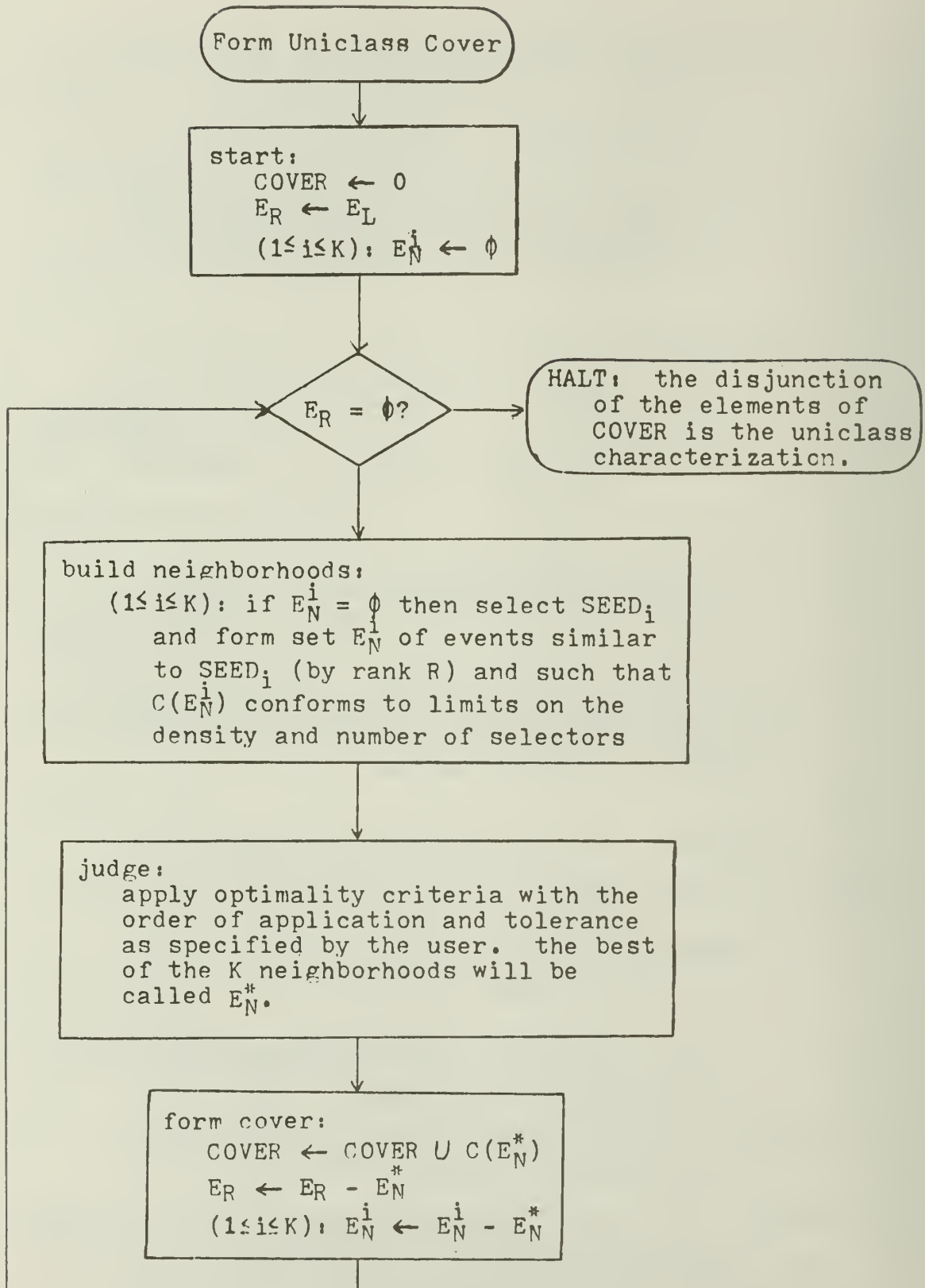
K - a user-specified number of neighborhoods to build in parallel and from which a "best neighborhood" is selected

$E_N^i$  - the set of events belonging to the ith neighborhood ( $1 \leq i \leq K$ )

SEED<sub>i</sub> - an event selected at random from set  $E_R$  which becomes the nucleus of neighborhood i (no two neighborhoods may have the same SEED)

COVER - a set of complexes of the best neighborhoods





Uniclass Algorithm  
Figure 1

The uniclass procedure begins when a set of events  $E_R = E_L$  has been established from user input data. The following steps are repeated until  $E_R = \emptyset$  and on each iteration another complex in the cover is produced, and the events which it covers are removed from  $E_R$ .

step 1: Build neighborhoods. A number (given by the user) of "neighborhoods" are constructed. A neighborhood is a set of events  $E_N$  such that for each event  $e'$  in  $E_N$ ,  $R(e', \text{seed})$  is not greater than a limit rank set by the user. "Seed" is an event selected arbitrarily from the set  $E_R$  and is unique to each neighborhood constructed.  $C(E_N)$  is the complex which covers the events  $E_N$ . The degree of generalization of  $C(E_N)$  is determined by the set  $E_N$ . During the neighborhood construction process some events are either excluded from  $E_N$  or forced into  $E_N$  in order to satisfy the user-given constraints of density threshold and/or selector threshold.

step 2: Select best neighborhood. A quasi-optimal neighborhood is selected according to one or more neighborhood judging criteria (ties are broken by making an arbitrary choice). Seven criteria are defined as follows.

- criterion 1: the number of complexes in the cover  
(estimated as the negative of the number  
of events in  $E_R$  covered by the complex)
- criterion 2: the number of selectors in a complex
- criterion 3: the sum of the costs of the variables  
in a formula (costs supplied by the user)
- criterion 4: the degree of generalization  
(estimated as  $1/D(C(E_N))$ )
- criterion 5: the sum of weights of the events covered  
by the complex (weights supplied by  
the user)
- criterion 6: the length of references ( $\sum_i s_i$ )
- criterion 7: the relative scope of references  
(the sum of the mean deviations for  
all variables)

These criteria are identical to the criteria available  
in program AQ7 [Larson & Michalski 75].

Neighborhood judging may be based on several criteria,  
applied in an ordering determined by the user. A  
tolerance value is specified for each and at each  
step of the judging, a neighborhood is eliminated  
if its criteria value is greater than an upper  
bound calculated as

$$UBOUND = MIN + TOLERANCE * (MAX - MIN).$$

step 3: Processing the chosen neighborhood. The best  
neighborhood represents a complex  $C(E_N)$  which will  
be saved to become one complex in the cover of the  
events. The events in  $E_N$  are eliminated from  $E_R$   
and from other neighborhoods which were not selected.



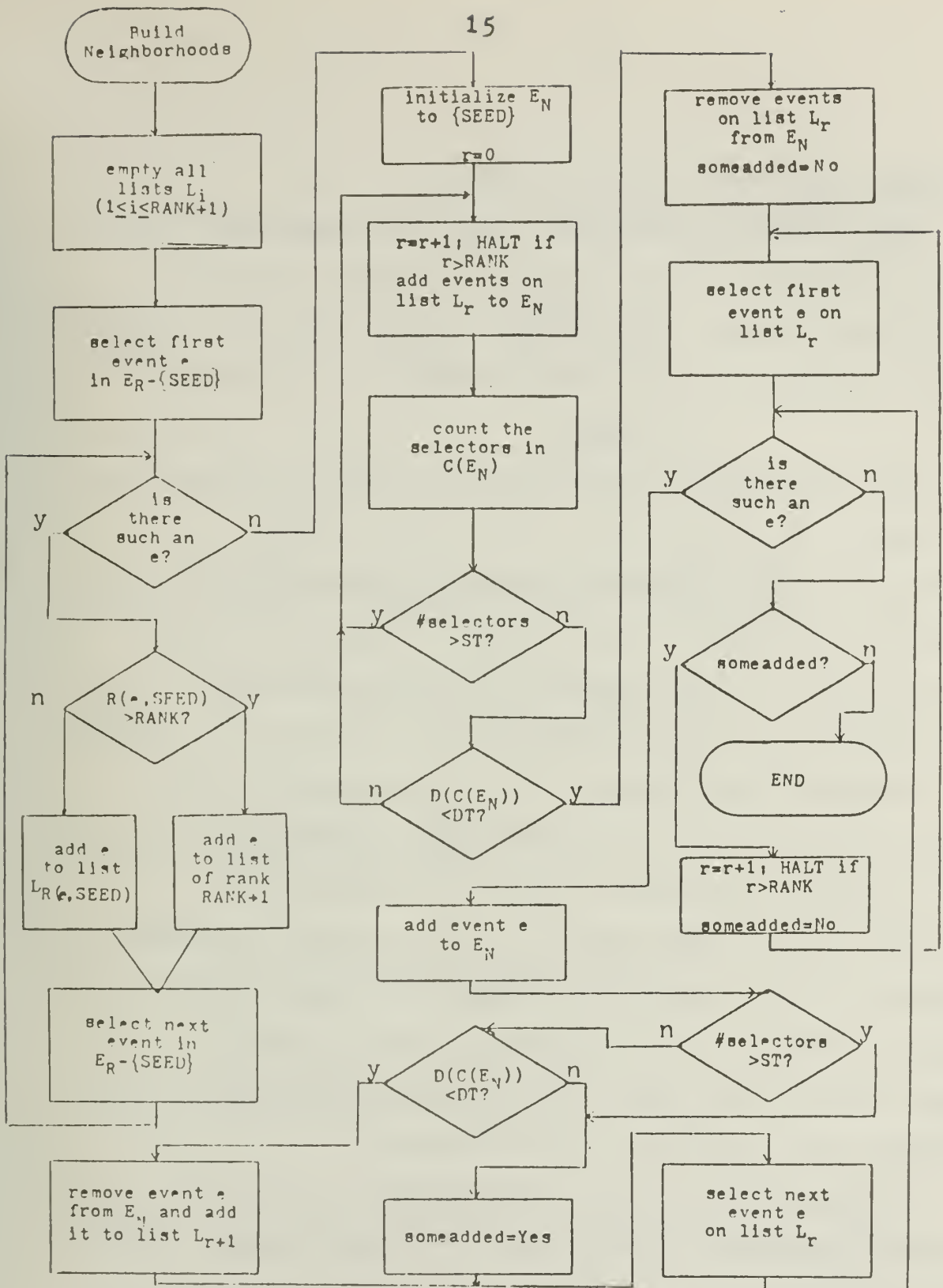
## Neighborhood Construction

Given a seed event, a rank limit, a density threshold, and a selector threshold, a neighborhood about the seed may be constructed. The resulting set of events  $E_N$  may be so small as to contain only the seed, or so large as to contain all of  $E_R$ . Controlling the size of  $E_N$  is important to achieving a useful characterization of the learning events. At any stage of the process, we may think of a neighborhood as a set of events  $E_N$  or as a complex  $C(E_N)$ . As  $E_N$  grows, it is possible that one or more selectors in  $C(E_N)$  will become such that its selector reference set will cover the entire domain of its associated variable. Such a selector is always true, and may be dropped from the complex. In this way, the number of selectors is linked to the arrangement of events in  $E_N$ .

To build a neighborhood about event seed, we first partition the set  $E_R$  into subsets  $E_{R1}, E_{R2}, \dots, E_{R\text{rank}}, E_{R\text{rank}+1}$ . All events in  $E_{Ri}$  are such that  $R(e, \text{seed})=i$  and all events in  $E_{R\text{rank}+1}$  are such that  $R(e, \text{seed}) > \text{rank}$ . ("rank" denotes the rank limit given by the user). The neighborhood construction algorithm attempts to form  $E_N$  as the union of the  $E_{Ri}$  sets for  $i$  from 1 to rank. Beginning as the set containing only the seed event,  $E_N$  has one subset  $E_{Ri}$  added to it, starting with  $E_{R1}$ . If

the selector threshold is satisfied (i.e. the number of selectors in  $C(E_N)$  at this point is not greater than the limit specified) then the density of  $C(E_N)$  is evaluated. If  $D(C(E_N))$  is not less than the density threshold, the process continues by going on to add events from the subset of  $E_R$  of next higher rank. If the density is too small, then the subset by subset construction process stops and an optional event by event construction process begins.

When the event by event construction process begins,  $E_N$  already contains the union of some  $E_{R1}$  subsets such that  $E_N$  satisfies the neighborhood constraints but the union of  $E_N$  with the subset of next higher rank does not. Let that subset of next higher rank be called  $E_{Rj}$ , then during event by event construction, each event in  $E_{Rj}$  is individually added to the set  $E_N$  temporarily and the desirability of  $E_N$  is evaluated. If  $E_N$  satisfies the neighborhood constraints then the individual event becomes a permanent part of  $E_N$ . Otherwise it is removed from  $E_N$  and placed instead into  $E_{Rj+1}$  where it is in a position to be considered again later. After all events in  $E_{Rj}$  have been considered neighborhood construction halts if none of the events in  $E_{Rj}$  were retained in  $E_N$ . If some events were retained, then the neighborhood construction continues to the next higher rank in the usual way.



## Legends:

SEED: an event which was selected from  $E_R$   
 $E_R$ : the set of events remaining to be covered  
 $L_i$ : a list of events whose distance to SEED (measured by  $R$ ) is  $i$   
RANK: a rank limit imposed by the user  
 $E_N$ : the set of events belonging to the neighborhood  
 $C(E_N)$ : the smallest single complex covering the events  $E_N$   
ST: the maximum-number-of-selectors threshold  
DT: the minimum-density threshold  
 $D(C)$ : the density of complex  $C$

Neighborhood Construction  
Figure 2

## Examples with Sample Characterizations

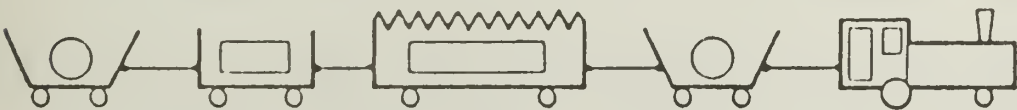
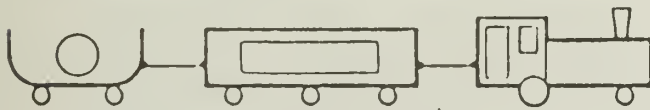
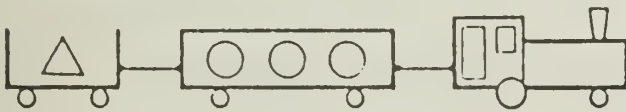
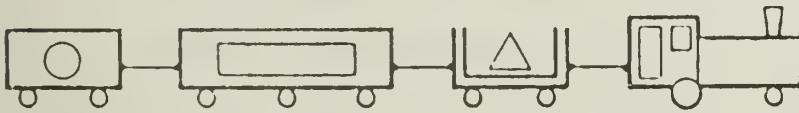
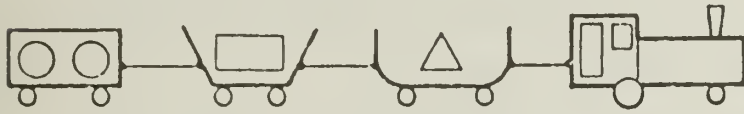
AQ7UNI will be illustrated using four examples from various articles on the VL<sub>1</sub> and VL<sub>2</sub> systems. The raw input specifications for all four examples are given in [Stepp 79] along with the actual output of the program.

### TRAINS

The first example is called TRAINS and it has been the subject of previous work using the VL<sub>2</sub> system [Larson 77]. The trains are presented pictorially in figure 3. In figure 3, two classes of trains are shown, however in the following characterization the trains are treated as one single class.

There are six domains in the trains example: the number of cars in the train, the number of wheels on car  $i$  ( $1 \leq i \leq 5$ ), the length of car  $i$ , the shape of car  $i$ , the shape of the cargo in car  $i$ , and the number of items carried in car  $i$ . When a train has fewer cars than the maximum number of 5, the parameters wheels, length, shape, cargo shape, number of items, for nonexistent cars are given the value "not applicable."

For this example it was decided that a characterization consisting of 2 complexes with approximately the same number of events in each was desirable. After several experimental choices of uniclass parameters,



TRAINS  
Figure 3  
from [Larson 77]

pleasing results were obtained using the degree of generalization given by a density threshold of  $10^{-6}$  with eight neighborhoods. An english translation of the complexes produced (see figure 4) is:

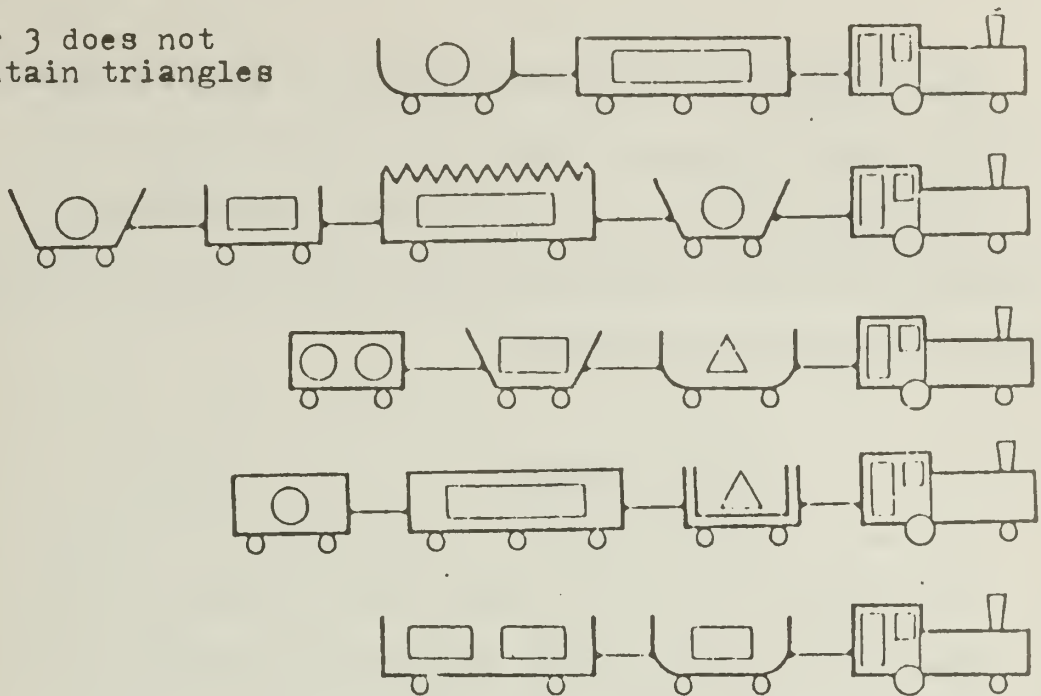
"A train is a sequence of five or fewer cars (this was presumed by the way in which the input data was set up) in which the first car is a locomotive (which is a long car having 2 wheels, with no cargo). The second car (the one attached to the locomotive) carries circles, triangles, or rectangles, and the fifth car (if it exists) is short, has 2 wheels, and carries 1 item. Additionally, there are two distinct types of trains. One type has circles or rectangles as cargo in car 3 while the other type has one triangle carried in car 3."

The complete output of the AQ7UNI program for the TRAINS problem can be found in [Stepp 79]. The program generates two complexes covering five trains each. The first part of the english translation above comes from a report in the output listing which gives the selectors which have identical values in both complexes, the "common characteristics." The statement that two types of trains exist reflects the two complexes, each describing five trains. The  $VL_1$  statement of these complexes (with common characteristics removed) is given in figure 4.



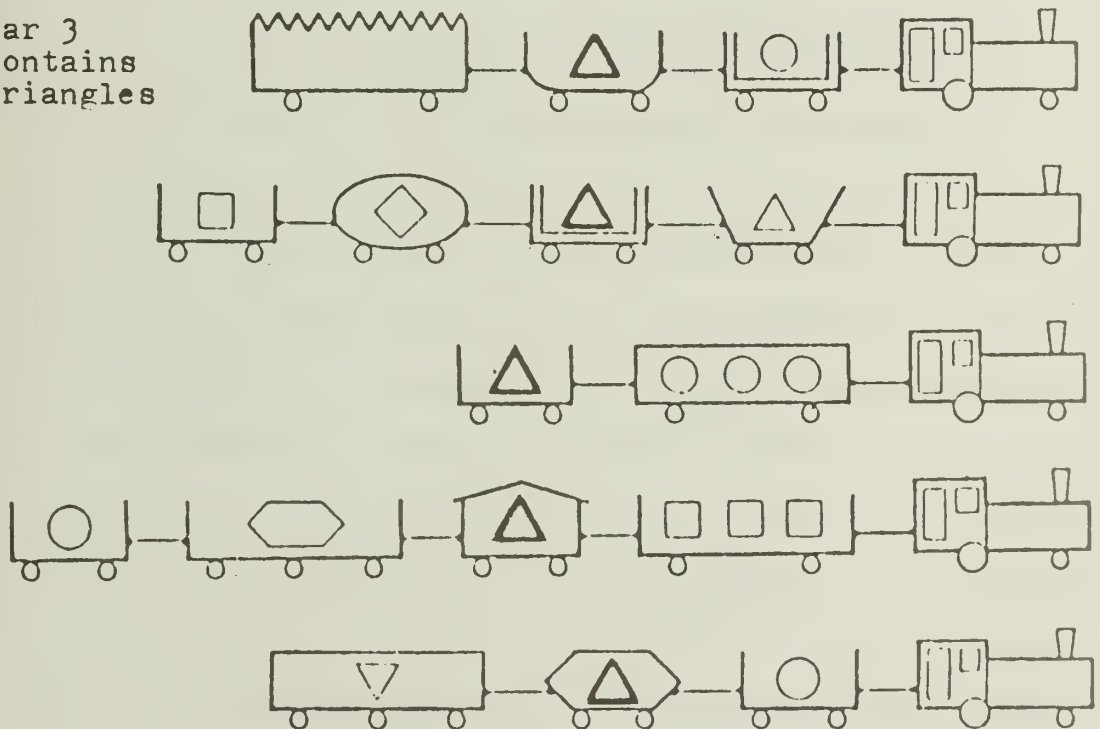
1

car 3 does not  
contain triangles



2

car 3  
contains  
triangles



COMPLEX 1 OF RANK 7 COVERS 5 EVENTS WITH DENSITY 2.17E-6  
(LSHAPE(3)=CIRCLE,RECTANGLE) (LOAD(2)=1) (LOAD(3)=1..2)

COMPLEX 2 OF RANK 10 COVERS 5 EVENTS WITH DENSITY 1.93E-6  
(LSHAPE(3)=TRIANGLE) (WHEELS(2)=2) (WHEELS(3)=2)  
(LENGTH(3)=SHORT) (LOAD(2)=1..3) (LOAD(3)=1)

Illustration of "TRAINS"

Figure 4

The trains are illustrated again in figure 4 where the two types of trains found by the characterization are shown by arranging the trains into two subgroups. A great many other characterizations are possible, using different uniclass parameters.

### BOTTLES

The wine bottles problem is illustrated in figure 5. In [Michalski 78] it is shown that bottles of wine produced by company A can be discriminated from those of company B Via the classification rules:

if [#circles=1] or  
[#triangles=1] then company A

if [#squares=1][#asterisks=1] then company B.

For our use here, all eight bottles are considered as one class. One characterization was produced with the selector threshold set to 2, and describes the bottles via three complexes which are illustrated in figure 6a. This characterization indicates that four bottles are marked with 1 square and 1 asterisk, three bottles are marked with 0 squares and 1 or 2 triangles while one unique bottle is marked with 1 square, 1 triangle, 0 circles, and 2 asterisks. It is interesting to note that even though this is a uniclass problem, one of the complexes (describing the top four bottles in figure 6a)



A:

1



2



3



4

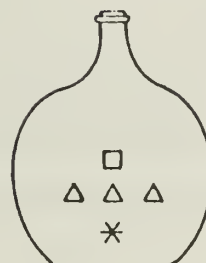


B:

1



2



3

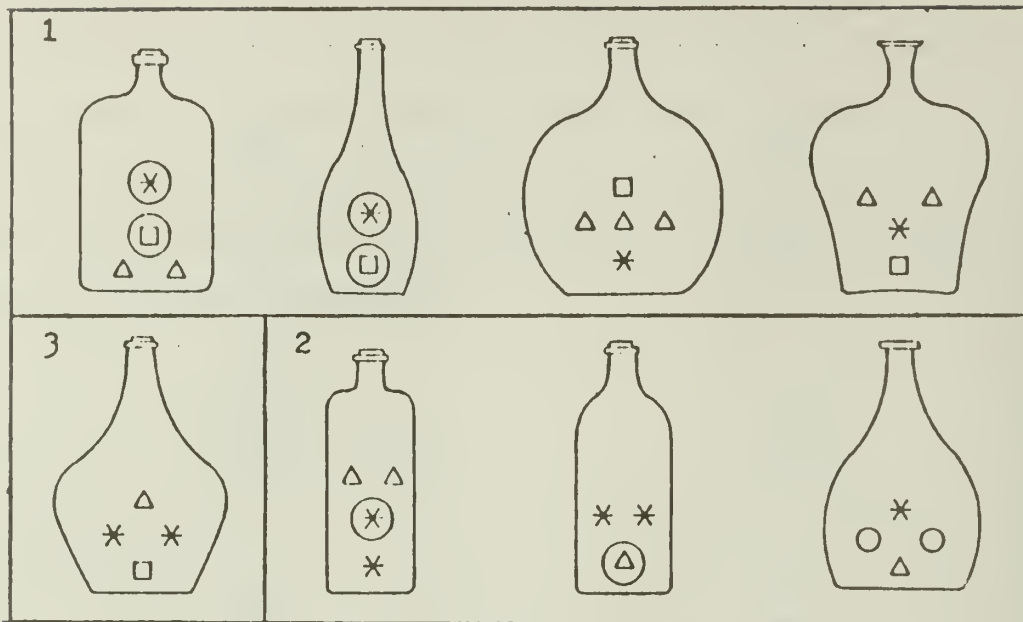


4



BOTTLES OF WINE  
Figure 5  
from [Michalski 78]

## Characterizations of BOTTLES

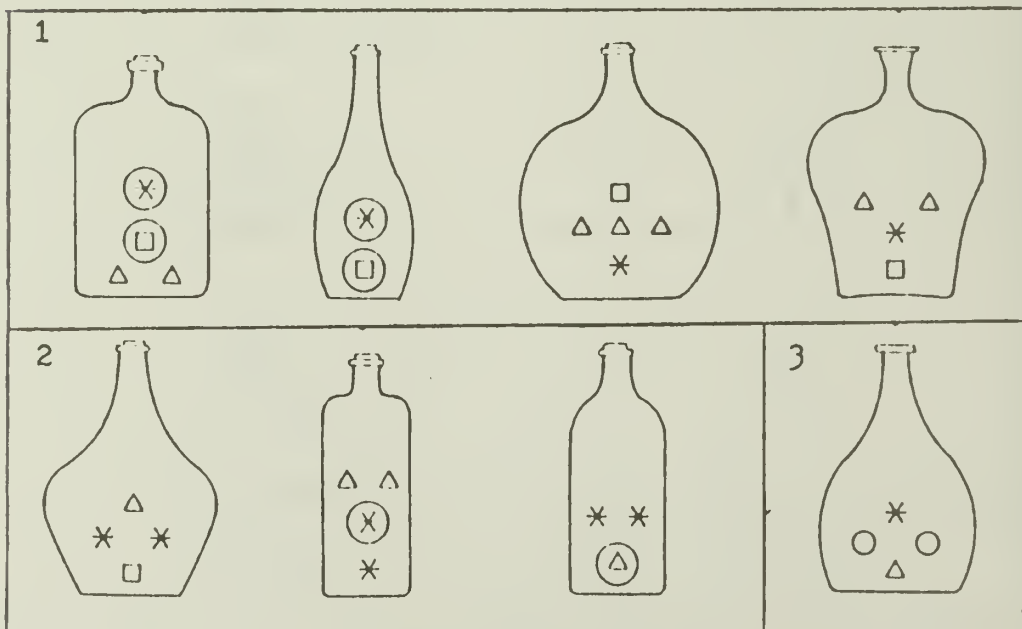


COMPLEX 1 OF RANK 2 COVERS 4 EVENTS ( 4 NEW) WITH DENSITY OF 3.333E-01  
 (#SQUARES=1) (#ASTERISKS=1)

COMPLEX 2 OF RANK 2 COVERS 3 EVENTS ( 3 NEW) WITH DENSITY OF 1.667E-01  
 (#SQUARES=0) (#TRIANGLES=1..2)

COMPLEX 3 OF RANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00  
 (#SQUARES=1) (#TRIANGLES=1) (#CIRCLES=0) (#ASTERISKS=2)

Figure 6a



COMPLEX 1 OF RANK 2 COVERS 4 EVENTS ( 4 NEW) WITH DENSITY OF 3.333E-01  
 (#SQUARES=1) (#ASTERISKS=1)

COMPLEX 2 OF RANK 3 COVERS 3 EVENTS ( 3 NEW) WITH DENSITY OF 3.750E-01  
 (#TRIANGLES=1..2) (#CIRCLES=0..1) (#ASTERISKS=2)

COMPLEX 3 OF RANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00  
 (#SQUARES=0) (#TRIANGLES=1) (#CIRCLES=2) (#ASTERISKS=1)

Figure 6b

describes precisely those bottles produced by company B.

Another characterization was made in which the degree of generalization was controlled by the density threshold (set to twice the overall event density) rather than the selector threshold. Figure 6b illustrates the three complexes in this second characterization. Surprisingly, the bottles produced by company B again form one of the complexes.

Figures 7a, 7b, and 7c show generalized logical diagrams (GLDs) for the three characterizations made. A generalized logical diagram contains a cell for each point in the event space. A complex is represented by the set of cells which represent the events the complex covers. Thus complexes are areas on the GLD. The characterizations of figures 6a and 6b correspond to the GLDs 7a and 7c respectively. These characterizations utilize disjoint complexes and this is clearly displayed by the GLDs, since no areas overlap. The program AQ7UNI can generate characterizations utilizing either disjoint or intersecting complexes at the user's option, and the characterization illustrated by GLD 7b was made with the same control parameters as in GLD 7a, except that intersecting complexes were used. GLDs are fully described in [Michalski 78].

## Generalized Logical Diagrams for BOTTLES

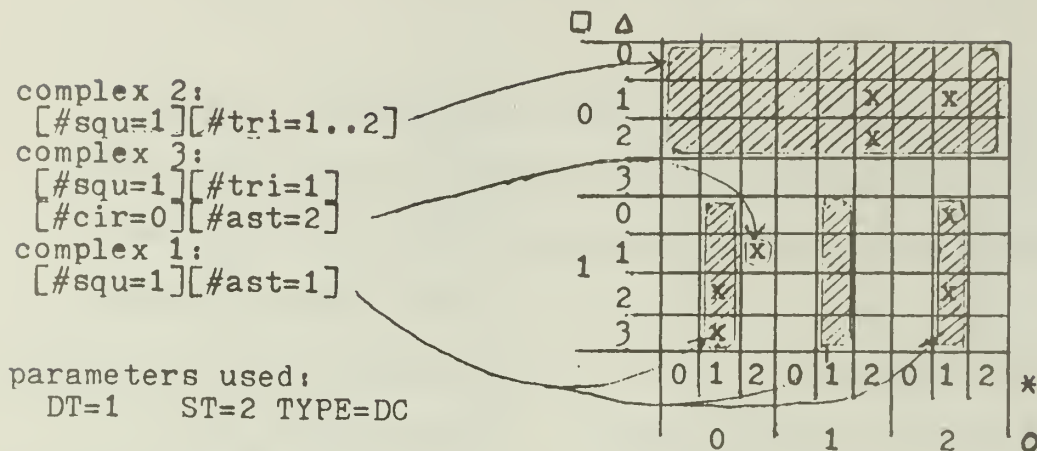


Figure 7a

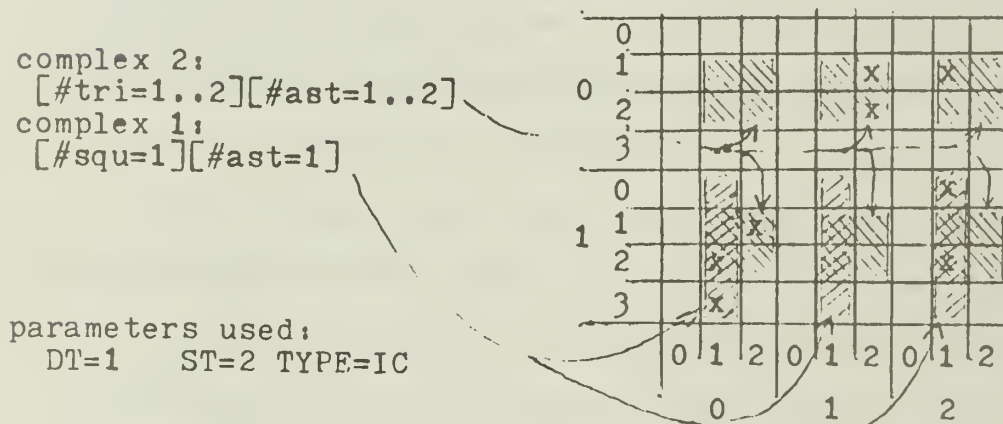


Figure 7b

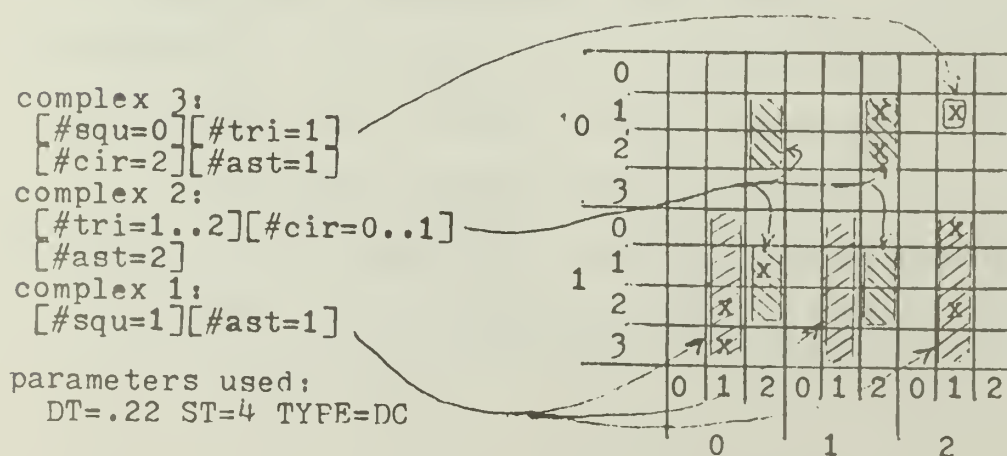


Figure 7c

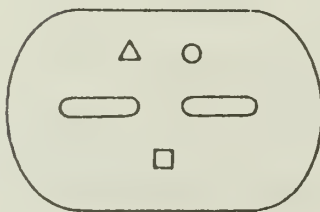
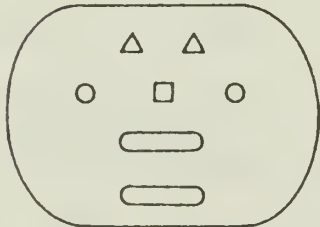
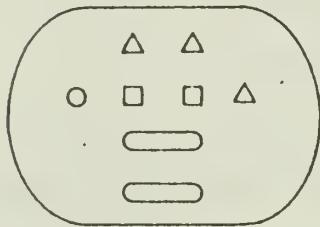
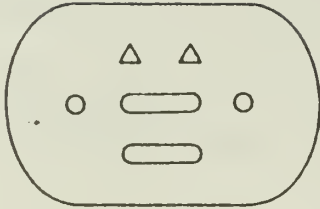
FACES

The next problem characterizes the "faces" presented in figure 8. Each face is described by the four features: number of circles, number of ovals, number of triangles, and number of squares. This example, like the others, started out as a discrimination problem and is treated as such in [Michalski 75]. For our use here all class boundaries are removed and all eight events are considered as a single class.

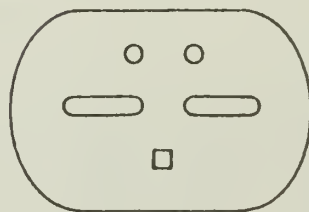
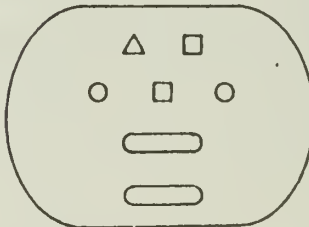
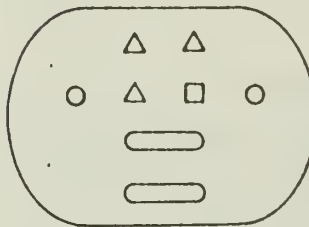
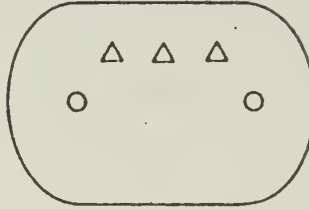
Two characterizations were produced. The first one is constrained via the selector threshold to use a maximum of three selectors in any complex, tending to cause some generalization to be made since four selectors are required to specify any single event. Additionally, the mode is specified as EXACT which limits the generalization to just that produced by the elimination of at least one selector. The first characterization is illustrated in figure 9a which shows the four complexes which cover 3, 2, 2, and 1 events respectively. Because these complexes are disjoint, the events they cover form clusters and in this case they are hierarchical. In figure 9a the horizontal line divides the eight faces according to the number of squares they have. The four faces above the line have 1 square while those below it have 0 or 2 squares. The top group is further divided according to the number of circles.

Characterization two in contrast to characterization

F(0)



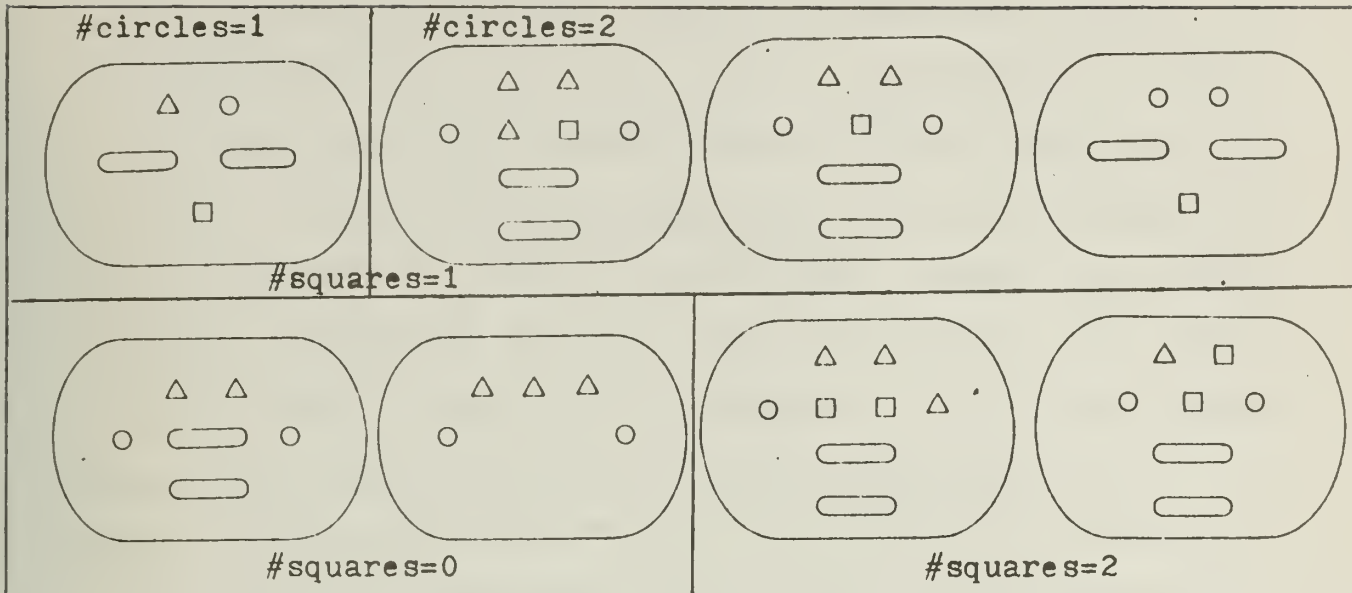
F(1)



FACES  
Figure 8

from [Michalski 75]

## Characterizations of FACES



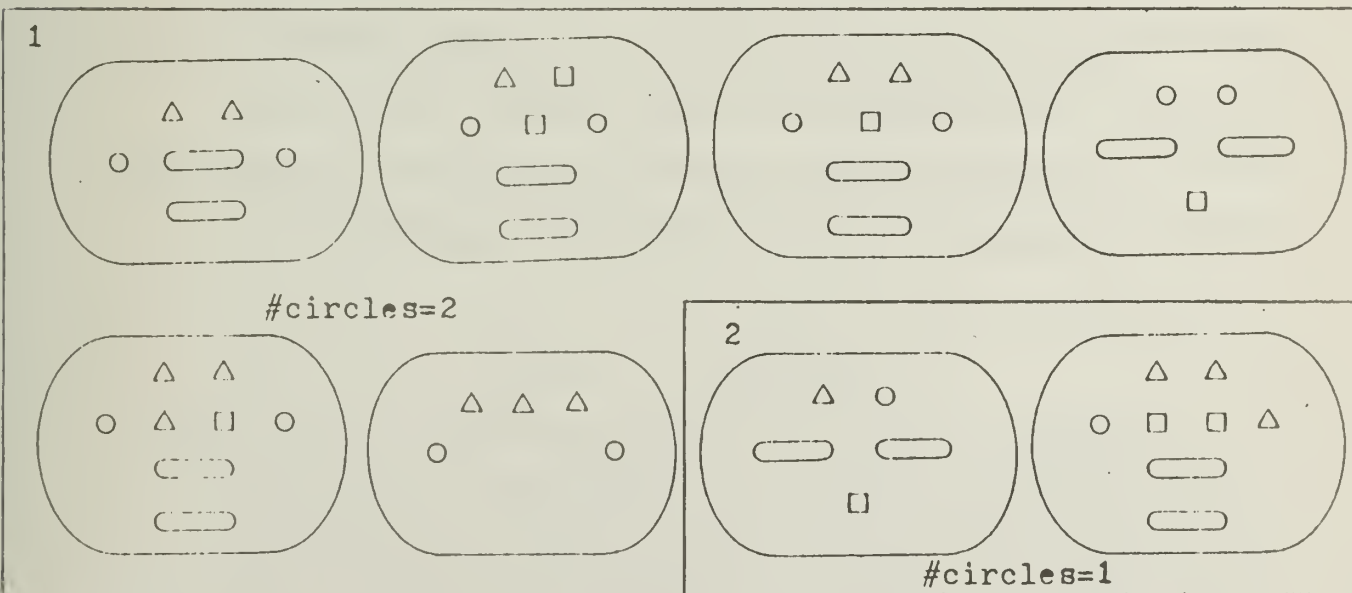
COMPLEX 1 OF BANK 1 COVERS 3 EVENTS ( 3 NEW) WITH DENSITY OF 7.5J0E-01  
 (#CIRCLES=2) (#OVALS=2) (#SQUARES=1)

COMPLEX 2 OF BANK 2 COVERS 2 EVENTS ( 2 NEW) WITH DENSITY OF 3.313E-01  
 (#CIRCLES=2) (#TRIANGLES=2..3) (#SQUARES=0)

COMPLEX 3 OF BANK 2 COVERS 2 EVENTS ( 2 NEW) WITH DENSITY OF 2.5J0E-01  
 (#CIRCLES=1..2) (#OVALS=2) (#SQUARES=2)

COMPLEX 4 OF BANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.JJ0E+00  
 (#CIRCLES=1) (#OVALS=2) (#TRIANGLES=1) (#SQUARES=1)

Figure 9a



COMPLEX 1 OF BANK 2 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 1.667E-01  
 (#CIRCLES=2)

COMPLEX 2 OF BANK 2 COVERS 2 EVENTS ( 2 NEW) WITH DENSITY OF 3.333E-01  
 (#CIRCLES=1) (#OVALS=2) (#TRIANGLES=1..3) (#SQUARES=1..2)

Figure 9b



one, is made with MODE set to FREE, which permits the greatest generalization, however the RANK of the complex has been limited to 2. The RANK value, when lower than the number of variables, saves time in the program by refusing to consider subgroupings of highly dissimilar events, i.e. the lower the RANK, the more similarity between events in the subgroups which a single complex will cover. The second characterization is illustrated in figure 9b. Now there are only two complexes, which divide the faces into two groups: those with 2 circles and those with 1 circle. The selection of the characterization of most utility rests with the user and depends on his current level of understanding of the environment in which the problem exists and the framework in which the characterization is to be used. By adjusting the parameters of the characterization, the cost, optimality, and degree of generalization may be varied widely. In the two characterizations presented, the length of references criterion was used to judge neighborhood optimality.

### ANIMALS

The animals example represents a family of problems in the biological sciences. A large number of animals, presumably of microscopic size, are shown in figure 10. This problem was taken from [Michalski 75] where it was used to illustrate the usefulness of Variable-Valued Logic to the classification problem. The features used



to describe the animals are the following.

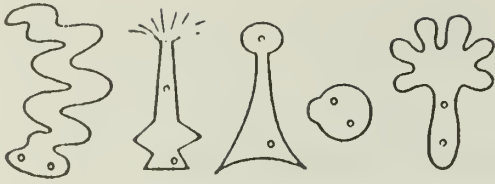
- number of black circles on the body
- number of tails
- number of crossmarks on the tails
- number of easily distinguished extremities
- type of body texture
- number of empty circles on the body
- number of empty squares on the body
- number of empty triangles on the body
- type of tail
- shape of body
- number of sharp or straight angles
- number of eyes
- number of black squares on the body

In generating characterizations of the animals, we will proceed in two ways:

- (1) A characterization of each class or phyla will be made separately. Since we assume animals within the same class are similar, we will seek a characterization of a high degree of generalization to establish common characteristics within each individual class.
- (2) A characterization of all animals will be made in which all classes shown in figure 10 are combined to produce just one class. Since we are now looking for similarities and differences among the animals, we will seek a characterization with a medium degree of generalization.

Output data for the ANIMALS example appears on pages to . There were 18 separate characterizations produced, which are listed starting on page 37. Of these, the first

## 0. JEXEMS:



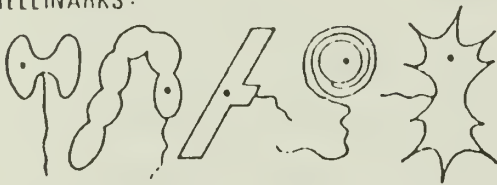
## 2. GRUFFLES:



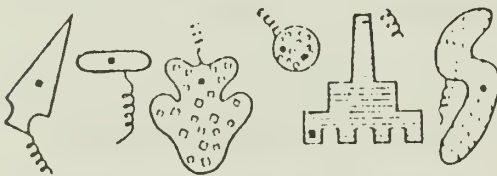
## 4. SNORPS:



## 6. MELLINARKS:



## 8. FUBBYLOOFERS:



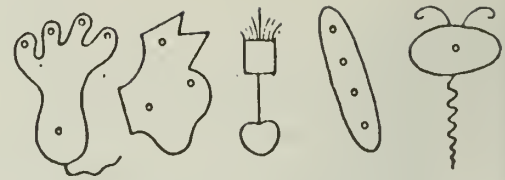
## 10. NORLEYS:



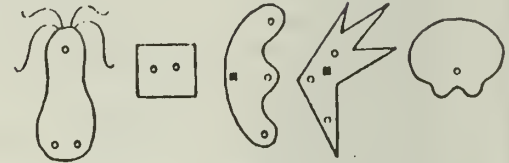
## 12. FLORGIEDORFLES:



## 1. SMUXEYS:



## 3. SELFFUNGS:



## 5. SPORONS:



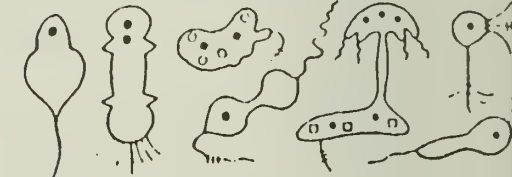
## 7. SCRANILLEMS:



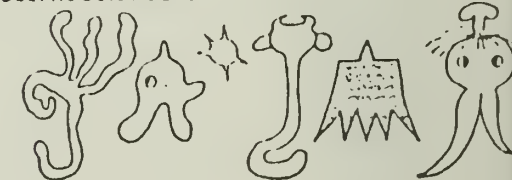
## 9. SREFOLYBUFFS:



## 11. SEYLRONS:



## 13. SELFROOEIGROLFS:



## SPECIES OF 'ANIMALS'

Figure 10

from [Michalski 75]

14 are individual characterizations of separate phyla, while the last 4 are characterizations of all animals. Looking at characterization number 1, which characterizes class 0, the Jexems, we see that these animals are ones which have no black circles, crossmarks, empty squares, empty triangles, eyes, black squares, or single tail but they do have two empty circles and are blank in texture and have irregular or circular shape. The individual phyla characterizations provide a good description of each class which, because it is mechanically generated, is always accurate.

Comparison of the characterizations of the individual phyla is also enlightening. When we compare characterization 3 (Gruffles) with characterization 1 (Jexems) we see that

Gruffles have no tails; Jexems may have tails.

Gruffles have 2 or more empty circles; Jexems have only 2 empty circles.

Gruffles always have empty triangles; Jexems never have any.

Gruffles may be ellipse shaped; Jexems may not be ellipse shaped.

Gruffles have no sharp angles; Jexems may have sharp angles.

These differences provide new information about the two classes which otherwise might go unnoticed. In the case presented here, Gruffles can be differentiated from Jexems

by checking the number of empty triangles feature. In general though, the characterizations of separate groups are not mutually disjoint, so discrimination rules cannot be produced merely by comparing the characterizations.

Now we consider an even more interesting characterization. Consider these animals in figure 10 before the phyla existed, and suppose that we believe it reasonable to make the phyla classifications according to the 13 features which have been defined. By requesting a characterization with a medium degree of generalization, we should expect to see all animals described via subgroups, in which the animals within each subgroup are similar. This was done and the results for two different characterizations are labeled characterizations 15 and 17 respectively in the program-generated output.

In characterization 15, the 8 complexes given divide the animals into 8 subgroups, which presumably could form the basis for new phyla classifications. In figure 11 the animals have been rearranged into these 8 subgroups. The largest subgroup contains the more "ordinary" animals. Whether the relatively large population of the first subgroup is good or bad depends on the user's viewpoint, however the uniclass algorithm always tends to cover the most events with the first complex when disjoint complexes are being produced, as in this case. The other 7 subgroups

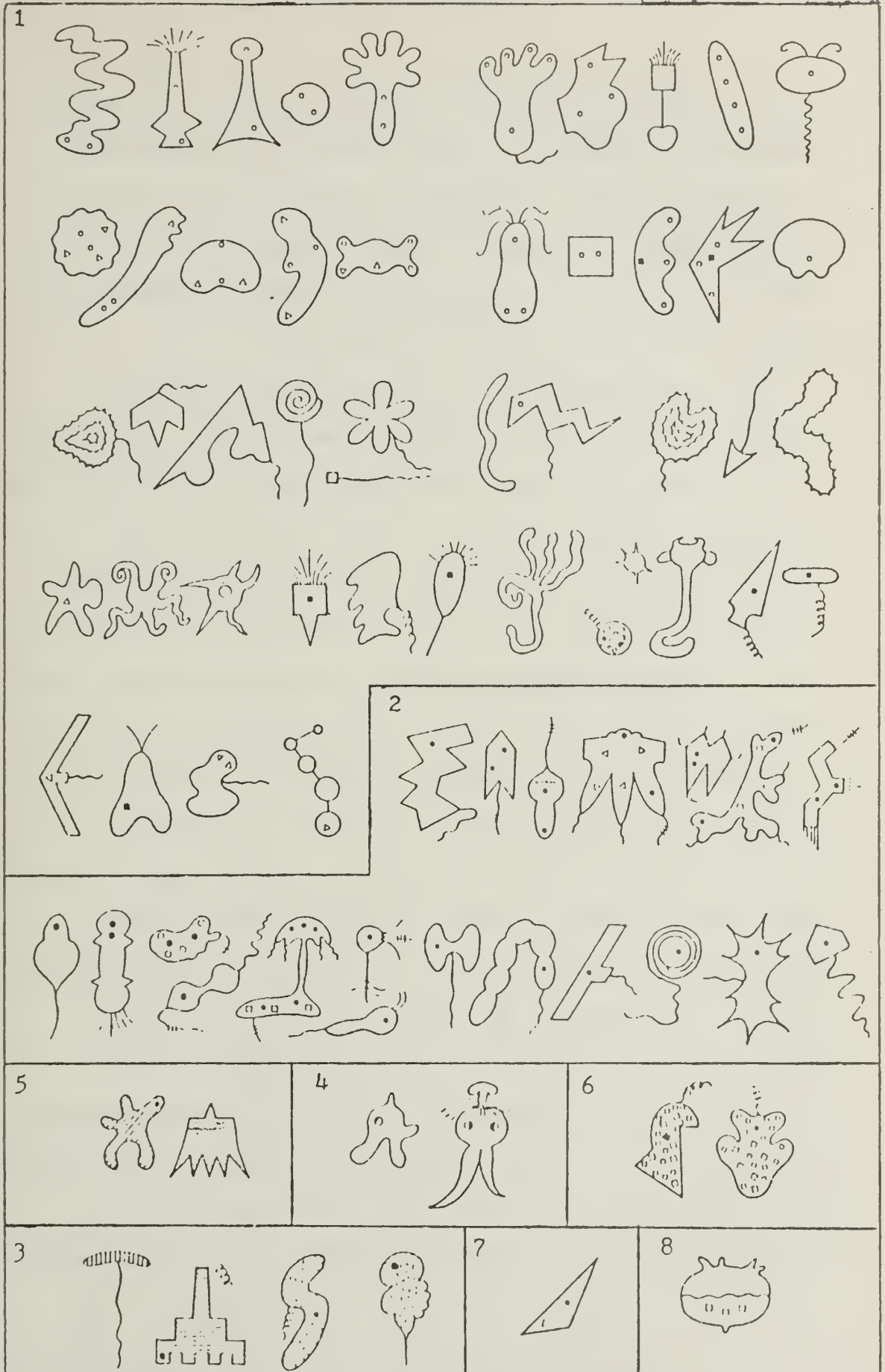


Figure 11: Characterization of ANIMALS

(see page 39 for VL<sub>1</sub> formulas)



contain animals of obvious similarity with the possible exception of subgroup 3. Even though the two animals in subgroup 3 look different, in terms of the descriptive features used to represent them, they are quite similar. Subgroups 7 and 8 show two special cases of animals with unique attributes relative to the others.

In characterization 17, the density threshold is slightly smaller and 12 complexes were formed. In figure 12 the animals are shown rearranged according to characterization 17. Because of the reduced generalization, the size of the first subgroup has reduced slightly. Even though many animals are grouped differently, the general nature of both characterizations 15 and 17 is essentially the same. This is in spite of the fact that the degree of generalization of characterization 15 is controlled by the selector threshold while in characterization 17 it is controlled by the density threshold. This similar behavior would tend to suggest that one of the thresholds is unnecessary and that the concept of degree of generalization is not fully captured by either of them. The author needed several attempts to choose uniclass parameters (which include other items as well as the two thresholds mentioned here) in order to create characterizations containing about ten complexes. There seems to be no reliable way to predict the number of complexes generated

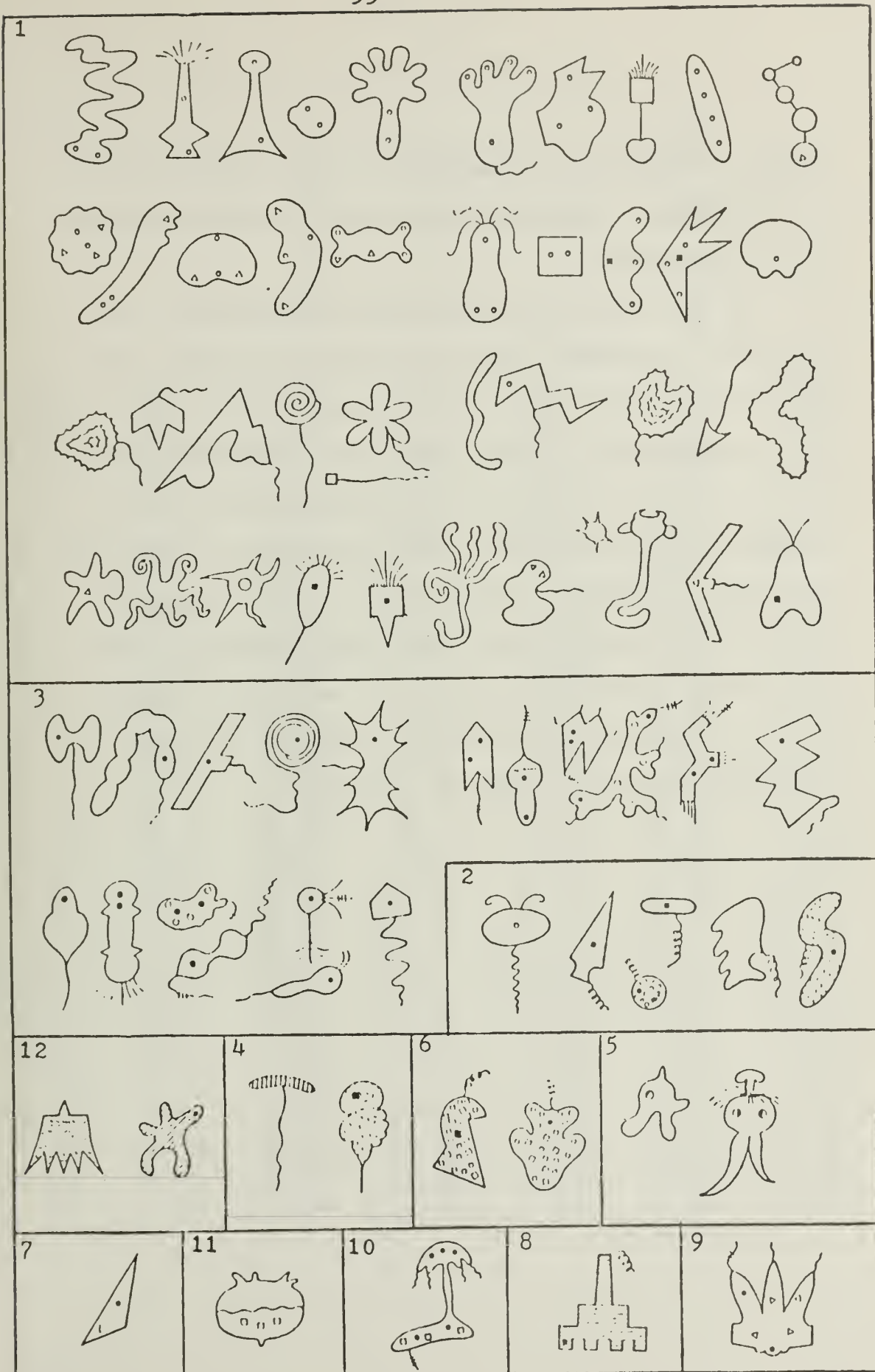


Figure 12: Characterization of ANIMALS

(see page 41 for VL<sub>1</sub> formulas)

from a certain set of parameters for a given problem specification. Experimentation is the only technique used at present.

In the two characterizations illustrated, the "disjoint complexes" mode was specified so that any individual animal is covered by exactly one complex and hence appears in just one subgroup. Characterizations 16 and 18 are identical to characterizations 15 and 17 respectively, except that the "intersecting complexes" mode was requested. Intersecting complexes provide simpler descriptions with more generalization, which may be well suited to descriptive analysis. Figures 11 and 12 present the taxonomy resulting from the "disjoint complexes" mode of operation.



VL<sub>1</sub> Characterizations of each Phyla

## #1 JEXEMS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 0  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 1.722E-05 DENSITY THRESHOLD IS 0.000E+00

COMPLEX 1 OF RANK 2 COVERS 5 EVENTS ( 5 NEW) WITH DENSITY OF 3.125E-01 AG=1.68  
 {BLK-CIRC=0} {#CROSSMRKS=0} {TEXTURE=BLANK} {#EMP-CIRC=2} {#EMP-SQ=0} {#EMP-TRIANG=0} {TAIL=NONE}  
 {SHAPE=IRREGULAR,CIRCLE} {#EYES=0} {#BLK-SQ=0}

## #2 SMUXEYS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 1  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 1.722E-05 DENSITY THRESHOLD IS 0.000E+00

COMPLEX 1 OF RANK 5 COVERS 5 EVENTS ( 5 NEW) WITH DENSITY OF 2.315E-02 AG=5.43  
 {BLK-CIRC=0} {#CROSSMRKS=0} {TEXTURE=BLANK} {#EMP-SQ=0} {#EMP-TRIANG=0} {SHAPE=IRREGULAR,ELLIPSE,TRIANG-SQ} {#EYES=0}  
 {#BLK-SQ=0}

## #3 GRUFFLES

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 2  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 1.722E-05 DENSITY THRESHOLD IS 0.000E+00

COMPLEX 1 OF RANK 3 COVERS 5 EVENTS ( 5 NEW) WITH DENSITY OF 4.167E-01 AG=1.26  
 {BLK-CIRC=0} {#TAILS=0} {#CROSSMRKS=0} {TEXTURE=BLANK} {#EMP-CIRC=2.3 OR MORE} {#EMP-SQ=0} {#EMP-TRIANG=1 OR MORE}  
 {TAIL=NONE} {SHAPE=IRREGULAR,ELLIPSE,CIRCLE} {#ANGLES=0} {#PIRS=0} {#BLK-SQ=0}

## #4 SELFFUNGS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 3  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 1.722E-05 DENSITY THRESHOLD IS 0.000E+00

COMPLEX 1 OF RANK 4 COVERS 5 EVENTS ( 5 NEW) WITH DENSITY OF 3.472E-02 AG=4.85  
 {BLK-CIRC=0} {#CROSSMRKS=0} {TEXTURE=BLANK} {#EMP-SQ=0} {#EMP-TRIANG=0} {SHAPE=IRREGULAR,ELLIPSE,TRIANG-SQ}  
 {#EYES=0}

## #5 SNORPS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 4  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.067E-05 DENSITY THRESHOLD IS 0.000E+00

COMPLEX 1 OF RANK 2 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 7.500E-01 AG=0.41  
 {BLK-CIRC=0} {#TAILS=1 OR MORE} {#CROSSMRKS=0} {TEXTURE=ELABK} {#EMP-CIRC=0 OR 1} {#EMP-SQ=0} {#EMP-TRIANG=0}  
 {TAIL=STRAIGHT} {SHAPE=IRREGULAR,TRIANG-SQ} {#EYES=0} {#BLK-SQ=0}

## #6 SPORONS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 5  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.067E-05 DENSITY THRESHOLD IS 0.000E+00

COMPLEX 1 OF RANK 4 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 1.875E-01 AG=2.42  
 {BLK-CIRC=0} {#CROSSMRKS=0} {#TAILS=0 OR 1} {TEXTURE=BLANK,VERT LINE} {#EMP-CIRC=0 OR 1} {#EMP-SQ=0} {#EMP-TRIANG=0}  
 {TAIL=NONE,STRAIGHT} {SHAPE=IRREGULAR,TRIANG-SQ} {#EYES=0} {#BLK-SQ=0}

## #7 MELLINARKS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 6  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 1.722E-05 DENSITY THRESHOLD IS 0.000E+00

COMPLEX 1 OF RANK 2 COVERS 5 EVENTS ( 5 NEW) WITH DENSITY OF 1.250E+00 AG=-.32  
 {BLK-CIRC=1} {#TAILS=1 OR MORE} {#CROSSMRKS=0} {TEXTURE=BLANK} {#EMP-CIRC=0 OR 1} {#EMP-SQ=0} {#EMP-TRIANG=0}  
 {TAIL=STRAIGHT} {SHAPE=IRREGULAR} {#EYES=0} {#BLK-SQ=0}

# VL1 Characterizations of each Phyla (continued)

## #8 SCRANILLEMS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 7  
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.067E-05 DENSITY THRESHOLD IS 0.000E+00  
COMPLEX 1 OF RANK 6 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 3.125E-02 AG=5.00  
{BLK-CIRC=0, 1} {CROSSHBKS=0} {TEXTURE=BLANK} {#EMP-CIRC=0 OR 1} {#EMP-SQ=0} {TAIL=NONE, STRAIGHT}  
{SHAPE=IRREGULAR, CIRCLE, TRIANG-SQ} {#EYES=0} {#BLK-SQ=0}

## #9 FUBBYLOOPERS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 8  
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.067E-05 DENSITY THRESHOLD IS 0.000E+00  
COMPLEX 1 OF RANK 3 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 4.167E-02 AG=4.58  
{BLK-CIRC=0} {#TAILS=1 OR MORE} {#CROSSHBKS=0} {#EXTREM=0 OR 1} {TEXTURE=BLANK, HORIZ LINE, COARSE} {#EMP-TRIANG=0}  
{TAIL=SPRING} {#EYES=0} {#BLK-SQ=1 OR MORE}

## #10 SREFOLYBUFFS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 9  
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.067E-05 DENSITY THRESHOLD IS 0.000E+00  
COMPLEX 1 OF RANK 3 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 4.167E-02 AG=4.58  
{BLK-CIRC=0} {#TAILS=1 OR MORE} {#CROSSHBKS=0} {#EXTREM=0 OR 1} {TEXTURE=BLANK, COARSE} {#EMP-CIRC=0 OR 1}  
{#EMP-TRIANG=0} {SHAPE=IRREGULAR, ELLIPSE, TRIANG-SQ} {#EYES=0}

## #11 NORLEYS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 10  
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.067E-05 DENSITY THRESHOLD IS 0.000E+00  
COMPLEX 1 OF RANK 5 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 6.250E-02 AG=4.00  
{BLK-CIRC=2 OR MORE} {#TAILS=1 OR MORE} {#CROSSHBKS=3} {TEXTURE=BLANK} {#EMP-SQ=0} {TAIL=NONE, STRAIGHT}  
{SHAPE=IRREGULAR, TRIANG-SQ} {#EYES=0} {#BLK-SQ=0}

## #12 SEYLONS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 11  
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.011E-05 DENSITY THRESHOLD IS 0.000E+00  
COMPLEX 1 OF RANK 4 COVERS 7 EVENTS ( 7 NEW) WITH DENSITY OF 4.861E-02 AG=4.36  
{BLK-CIRC=1-2 OR MORE} {#TAILS=1 OR MORE} {#EXTREM=0 OR 1} {TEXTURE=BLANK} {#EMP-TRIANG=0}  
{SHAPE=IRREGULAR} {#EYES=0} {#BLK-SQ=0} {TAIL=NONE, STRAIGHT}

## #13 FLORGIEDORFLES

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 12  
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 1.722E-05 DENSITY THRESHOLD IS 0.000E+00  
COMPLEX 1 OF RANK 3 COVERS 5 EVENTS ( 5 NEW) WITH DENSITY OF 3.125E-01 AG=1.68  
{BLK-CIRC=0-1} {#TAILS=0} {#CROSSHBKS=0} {#EXTREM=2 OR MORE} {TEXTURE=BLANK, DIAG LINES} {#EMP-CIRC=0 OR 1}  
{TAIL=NONE} {SHAPE=IRREGULAR} {#ANGLES=0} {#EYES=0} {#BLK-SQ=0}

## #14 SELFRODEIGROLFS

THE FOLLOWING 1 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 13  
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.067E-05 DENSITY THRESHOLD IS 0.000E+00  
COMPLEX 1 OF RANK 2 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 7.500E-01 AG=0.41  
{BLK-CIRC=0} {#TAILS=0} {#CROSSHBKS=0} {#EXTREM=2 OR MORE} {TEXTURE=BLANK, WAVES} {#EMP-CIRC=0 OR 1} {#EMP-SQ=0}  
{#EMP-TRIANG=0} {TAIL=NONE} {SHAPE=IRREGULAR} {#BLK-SQ=0}



(16) MODE = PRZE, 1 DT = 0.00000000, ST = 8 0.00000000, RANK = 3 TYPE = IC ULIST = 111111111111111111

THE FOLLOWING 7 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 3 1 1/2 3 4 5 6 7 8 9 10  
 11 12 13  
 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.721E-04 DENSITY THRESHOLD IS 0.0002+00  
 COMPLEX 1 OF RANK 3 COVERS 47 EVENTS ( 47 NEW) WITH DENSITY OF 4.080E-02 AG=4.62  
 (BLK-CIRC=0) (#CROSSMARKS=0) (TEXTURE=BLANK) (#EMP-SQ=0) (#EYES=0)  
 COMPLEX 2 OF RANK 3 COVERS 54 EVENTS ( 12 NEW) WITH DENSITY OF 8.789E-03 AG=6.83  
 (BLK-CIRC=0..1) (#CROSSMARKS=0) (TEXTURE=BLANK) (TAIL=NONE, STRAIGHT)  
 COMPLEX 3 OF RANK 3 COVERS 64 EVENTS ( 12 NEW) WITH DENSITY OF 4.115E-03 AG=7.92  
 (TEXTURE=BLANK, VERT LINES) (#EMP-SQ=0) (SHAPE=IRREGULAR, ELLIPSE, TRIANG-SQ) (#EYES=0)  
 COMPLEX 4 OF RANK 3 COVERS 58 EVENTS ( 2 NEW) WITH DENSITY OF 4.196E-03 AG=7.90  
 (BLK-CIRC=0..1) (#CROSSMARKS=0) (TEXTURE=BLANK, VERT LINES) (SHAPE=IRREGULAR, ELLIPSE, TRIANG-SQ)  
 COMPLEX 5 OF RANK 3 COVERS 17 EVENTS ( 3 NEW) WITH DENSITY OF 5.060E-02 AG=4.30  
 (BLK-CIRC=0) (#TAILS=1 OR MORE) (#CROSSMARKS=0) (#EXTREM=0 OR 1) (#EMP-CIRC=0 OR 1) (#EMP-TRIANG=0)  
 (SHAPE=IRREGULAR, TRIANG-SQ) (#EYES=0)  
 COMPLEX 6 OF RANK 3 COVERS 7 EVENTS ( 2 NEW) WITH DENSITY OF 1.667E-01 AG=2.58  
 (TAILS=0) (#CROSSMARKS=0) (#EXTREM=2 OR MORE) (#EMP-CIRC=0 OR 1) (#EMP-SQ=0) (#EMP-TRIANG=0) (TAIL=NONE)  
 (SHAPE=IRREGULAR) (#EYES=0) (BLK-SQ=0)  
 COMPLEX 7 OF RANK 3 COVERS 10 EVENTS ( 1 NEW) WITH DENSITY OF 5.208E-02 AG=4.26  
 (BLK-CIRC=1..2 OR MORE) (#TAILS=1 OR MORE) (#CROSSMARKS=3) (TEXTURE=BLANK) (#EMP-CIRC=0 OR 1)  
 (SHAPE=IRREGULAR, TRIANG-SQ) (#EYES=0) (#BLK-SQ=0)

THE FOLLOWING SELECTORS ARE COMMON CHARACTERISTICS  
 ( NONE )

# VL<sub>1</sub> Characterizations of All Animals (continued) (intersecting complexes)



(17) MODK = APPROX 1 DT = 5.000E-02 ST = 13 RGB = 6 BANK = 3 TYPE = DC ULIST = 11111111111111  
 CLIST = 6 TLIST = 0.000 0.000

THE FOLLOWING 12 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 0 1 2 3 4 5 6 7 8 9 10  
 11 DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.721E-04 DENSITY THRESHOLD IS 5.000E-02

COMPLEX 1 OF BANK 3 COVERS 42 EVENTS ( 42 NEW) WITH DENSITY OF 5.469E-02 AG=4.19  
 (BLK-CIRC=0) (#CROSSMRKS=0) (TEXTURE=BLANK) (#EMP-SQ=0) (TAIL=NONE, STRAIGHT) (#YES=0)

COMPLEX 2 OF BANK 3 COVERS 6 EVENTS ( 6 NEW) WITH DENSITY OF 6.250E-02 AG=4.00  
 (BLK-CIRC=0) (#TAILS=1 OR MORE) (#CROSSMRKS=0) (#EXTREM=0 OR 1) (TEXTURE=BLANK, COARSE) (#EMP-SQ=0) (#EMP-TRIANG=0)  
 (TAIL=SPRING) (#YES=0)

COMPLEX 3 OF BANK 3 COVERS 18 EVENTS ( 18 NEW) WITH DENSITY OF 6.250E-02 AG=4.00  
 (BLK-CIRC=0) (#TAILS=1 OR MORE) (TEXTURE=BLANK) (#EMP-SQ=0) (TAIL=NONE, STRAIGHT)  
 (SHAPE=IRREGULAR, TRIANG-SQ=0) (#YES=0) (BLK-SQ=0)

COMPLEX 4 OF BANK 3 COVERS 2 EVENTS ( 2 NEW) WITH DENSITY OF 2.500E-01 AG=2.00  
 (BLK-CIRC=0) (#TAILS=1 OR MORE) (#CROSSMRKS=0) (#EXTREM=0 OR 1) (TEXTURE=COARSE, VERT LINES) (#EMP-CIRC=0 OR 1)  
 (#EMP-SQ=0) (#EMP-TRIANG=0) (TAIL=STRAIGHT) (SHAPE=IRREGULAR) (#YES=0)

COMPLEX 5 OF BANK 1 COVERS 2 EVENTS ( 2 NEW) WITH DENSITY OF 1.000E+00 AG=0.00  
 (BLK-CIRC=0) (#TAILS=0) (#CROSSMRKS=0) (#EXTREM=2 OR MORE) (TEXTURE=BLANK) (#EMP-CIRC=0 OR 1) (#EMP-SQ=0)  
 (#EMP-TRIANG=0) (TAIL=NONE) (SHAPE=IRREGULAR) (#YES=1 OR MORE) (#BLK-SQ=0)

COMPLEX 6 OF BANK 1 COVERS 2 EVENTS ( 2 NEW) WITH DENSITY OF 1.000E+00 AG=0.00  
 (BLK-CIRC=0) (#TAILS=1 OR MORE) (#CROSSMRKS=0) (#EXTREM=0 OR 1) (TEXTURE=BLANK) (#EMP-CIRC=0 OR 1) (#EMP-SQ=1 OR MORE)  
 (#EMP-TRIANG=0) (TAIL=SPRING) (SHAPE=IRREGULAR) (#YES=0) (BLK-SQ=1 OR MORE)

COMPLEX 7 OF BANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00 AG=0.00  
 (BLK-CIRC=1) (#TAILS=0) (#CROSSMRKS=0) (#EXTREM=0 OR 1) (TEXTURE=BLANK) (#EMP-CIRC=0 OR 1) (#EMP-SQ=0)  
 (TAIL=NONE) (SHAPE=TRIANG-SQ) (ANGLES=1 OR MORE) (#YES=0) (BLK-SQ=0)

COMPLEX 8 OF BANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00 AG=0.00  
 (BLK-CIRC=0) (#TAILS=1 OR MORE) (#CROSSMRKS=0) (#EXTREM=0 OR 1) (TEXTURE=HORIZ LINE) (#EMP-CIRC=0 OR 1) (#EMP-SQ=0)  
 (#EMP-TRIANG=0) (TAIL=SPRING) (SHAPE=TRIANG-SQ) (#ANGLES=1 OR MORE) (#YES=0) (#BLK-SQ=1 OR MORE)

COMPLEX 9 OF BANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00 AG=0.00  
 (BLK-CIRC=2 OR MORE) (#TAILS=1 OR MORE) (#CROSSMRKS=3) (#EXTREM=2 OR MORE) (TEXTURE=BLANK) (#EMP-CIRC=0 OR 1)  
 (#EMP-SQ=0) (#EMP-TRIANG=1 OR MORE) (TAIL=NONE) (SHAPE=IRREGULAR) (#ANGLES=1 OR MORE) (#YES=0) (#BLK-SQ=0)

COMPLEX 10 OF BANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00 AG=0.00  
 (BLK-CIRC=2 OR MORE) (#TAILS=1 OR MORE) (#CROSSMRKS=3) (#EXTREM=0 OR 1) (TEXTURE=BLANK) (#EMP-CIRC=0 OR 1)  
 (#EMP-SQ=1 OR MORE) (#EMP-TRIANG=0) (TAIL=NONE) (SHAPE=IRREGULAR) (#ANGLES=0) (#YES=0) (BLK-SQ=0)

COMPLEX 11 OF BANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00 AG=0.00  
 (BLK-CIRC=0) (#TAILS=0) (#CROSSMRKS=0) (#EXTREM=2 OR MORE) (TEXTURE=BLANK) (#EMP-CIRC=0 OR 1) (#EMP-SQ=1 OR MORE)  
 (#EMP-TRIANG=0) (TAIL=NONE) (SHAPE=IRREGULAR) (#ANGLES=0) (#YES=0) (BLK-SQ=0)

COMPLEX 12 OF BANK 3 COVERS 2 EVENTS ( 2 NEW) WITH DENSITY OF 2.500E-01 AG=2.00  
 (BLK-CIRC=0-1) (#TAILS=0) (#CROSSMRKS=0) (#EXTREM=2 OR MORE) (TEXTURE=WAVES, DIAG LINES) (#EMP-CIRC=0 OR 1)  
 (#EMP-SQ=0) (#EMP-TRIANG=0) (TAIL=NONE) (SHAPE=IRREGULAR) (#YES=0) (#BLK-SQ=0)

THE FOLLOWING SELECTORS ARE COMMON CHARACTERISTICS  
 ( NONE )

VL1 Characterizations of All Animals (continued)  
 (this characterization is illustrated in figure 12)  
 (disjoint complexes)

```

(18)  MODE = APPROX      DT = 5.000E-02      ST = 13      MGB = 6      MARK = 3      TYPE = IC      ULIST = 11111111111111
      CLIST = 6          TLIST = 0.000

THE FOLLOWING 10 CARTESIAN COMPLEXES FORM THE UNICLASS COVER FOR CLASSES 0 1 2 3 4 5 6 7 8 9 10
11 12 13
DENSITY OF LEARNING EVENTS IN EVENT SPACE IS 2.721E-04 DENSITY THRESHOLD IS 5.000E-02

COMPLEX 1 OF BANK 3 COVERS 42 EVENTS ( 42 NEW) WITH DENSITY OF 5.469E-02 AG=4.19
(BLK-CIRC=0) (#CROSSMARKS=0) (TEXTURE=BLANK) (#EXP-SQ=0) (TAIL=NONE, STRAIGHT) (#EYES=0)

COMPLEX 2 OF BANK 3 COVERS 39 EVENTS ( 2 NEW) WITH DENSITY OF 5.078E-02 AG=4.30
(BLK-CIRC=0) (#CROSSMARKS=0) (TEXTURE=BLANK) (#EXP-SQ=0) (TAIL=NONE, STRAIGHT) (#BLK-SQ=0)

COMPLEX 3 OF BANK 3 COVERS 42 EVENTS ( 9 NEW) WITH DENSITY OF 5.469E-02 AG=4.19
(BLK-CIRC=0..1) (#CROSSMARKS=0) (TEXTURE=BLANK) (#EXP-SQ=0) (#EMP-TRIANG=0) (TAIL=NONE, STRAIGHT) (#EYES=0)

COMPLEX 4 OF BANK 3 COVERS 34 EVENTS ( 2 NEW) WITH DENSITY OF 5.903E-02 AG=4.08
(BLK-CIRC=0..1) (#CROSSMARKS=0) (TEXTURE=BLANK, VERT LINES) (#EMP-SQ=0) (SHAPE=IRREGULAR) (#EYES=0) (#BLK-SQ=0)

COMPLEX 5 OF BANK 3 COVERS 30 EVENTS ( 9 NEW) WITH DENSITY OF 5.208E-02 AG=4.26
(BLK-CIRC=1..2 OR MORE) (TAILS=1 OR MORE) (#EXTREM=0 OR 1) (TEXTURE=BLANK) (#EMP-TRIANG=0) (TAIL=NONE, STRAIGHT)
(SHAPE=IRREGULAR, TRIANG-SQ) (#EYES=0) (#BLK-SQ=0)

COMPLEX 6 OF BANK 3 COVERS 15 EVENTS ( 2 NEW) WITH DENSITY OF 5.208E-02 AG=4.26
(BLK-CIRC=0) (TAILS=1 OR MORE) (#EXTREM=0 OR 1) (TEXTURE=BLANK) (#EMP-TRIANG=0) (TAIL=NONE, STRAIGHT)
(SHAPE=IRREGULAR, TRIANG-SQ) (#EYES=0) (#BLK-SQ=0)

COMPLEX 7 OF BANK 3 COVERS 5 EVENTS ( 5 NEW) WITH DENSITY OF 6.944E-02 AG=3.85
(BLK-CIRC=0) (TAILS=1 OR MORE) (#CROSSMARKS=0) (#EXTREM=0 OR 1) (TEXTURE=BLANK, COARSE) (#EMP-TRIANG=0) (TAIL=SPRING)
(SHAPE=IRREGULAR, ELLIPSE, CIRCLE) (#ANGLES=0) (#EYES=0)

COMPLEX 8 OF BANK 0 COVERS 1 EVENTS ( 1 NEW) WITH DENSITY OF 1.000E+00 AG=0.00
(BLK-CIRC=2 OR MORE) (TAILS=1 OR MORE) (#CROSSMARKS=3) (#EXTREM=2 OR MORE) (TEXTURE=BLANK) (#EMP-SQ=0) (#EMP-TRIANG=0) (TAIL=NONE) (SHAPE=IRREGULAR) (#ANGLES=0) (#EYES=0) (#BLK-SQ=0)

COMPLEX 9 OF BANK 3 COVERS 8 EVENTS ( 3 NEW) WITH DENSITY OF 3.333E-01 AG=1.59
(BLK-CIRC=0..1) (TAILS=0) (#CROSSMARKS=0) (#EXTREM=2 OR MORE) (TEXTURE=BLANK, WAVES, DIAG LINES) (#EMP-CIRC=0 OR 1)
(#EMP-TRIANG=0) (TAIL=NONE) (SHAPE=IRREGULAR) (#EYES=0) (#BLK-SQ=0)

COMPLEX 10 OF BANK 3 COVERS 6 EVENTS ( 4 NEW) WITH DENSITY OF 1.250E-01 AG=3.00
(BLK-CIRC=0) (TAILS=1 OR MORE) (#CROSSMARKS=0) (#EXTREM=0 OR 1) (TEXTURE=BLANK, HORIZ LINE, COARSE) (#EMP-CIRC=0 OR 1)
(#EMP-TRIANG=0) (TAIL=STRAIGHT, SPIRING) (SHAPE=IRREGULAR, TRIANG-SQ) (#EYES=0) (#BLK-SQ=1 OR MORE)

```

THE FOLLOWING SELECTORS ARE COMMON CHARACTERISTICS  
( NONE )

## VL<sub>1</sub> Characterizations of All Animals (continued) (intersecting complexes)

### Comparison with other Methods

The uniclass algorithm used by AQ7UNI differs from other characterization techniques in that

- (1) the uniclass algorithm can produce a disjunctive description of the events with varying degrees of generality, and
- (2) the uniclass algorithm does not permit the use of structured events (i.e. event descriptions involving dummy variables).

AQ7UNI is a data-driven, bottom-up method (as opposed to a top-down or model-driven method). The disjunctive units, the complexes, are built up of individual events until a threshold limit causes this process to halt. The characterization methods of SPROUTER [Hayes-Roth 76] and THOTH [Vere 78] have been classified as bottom-up methods in [Dietterich & Michalski 79] but AQ7UNI differs widely from both according to the two points above.

Even with the differences that exist, some generalization techniques do appear in common. In AQ7UNI, generalizations come about by the application of several procedures:

#### 1. internal disjunction

Recall that a selector is of the form  $[X_i = \text{value}]$  or  $[X_i = \text{set of values}]$ . The latter form represents an internal disjunction, i.e.  $X_2 = 3, 5$  represents



$(X_2=3) \vee (X_2=5).$

2. dropping a selector

When the list of values in a selector contains all values in the domain of the variable, or when a large portion of the domain is present and the action of the selector threshold forces the elimination of a selector, the selector is dropped from the characterization.

3. closing an interval

When variable  $X_i$  is of interval type and a selector such as  $X_i=2,5,8$  is present, it is generalized to  $X_i=2..8$  which denotes that  $X_i$  may take any value in the interval  $[2,8]$ .

4. climbing a generalization tree

When variable  $X_j$  is of structure type and a selector such as  $X_j=\text{square,triangle}$  is present, it is generalized by replacing the values by the term for which they are both refinements. In this example if square and triangle are both refinements of polygon, then the selector would become  $X_j=\text{polygon}$ . When no node of common refinement exists, the selector is dropped.

In a comparative study of several characterization methods [Dietterich & Michalski 79] these same generalization processes were found. Process number 2 is used in both

bottom-up methods mentioned previously. Processes 1 and 2 are incorporated in Meta-DENDRAL [Buchanan 78] and all processes are present in INDUCE [Larson 77]. These latter two programs utilize a model-driven technique.

It is important to remember that the characterization methods mentioned above utilize structured event environments. This capability is not supported by the VL<sub>1</sub> system, and hence is outside the realm of AQ7UNI. The importance of structured events can be illustrated by trying to solve the characterization problem used in [Dietterich & Michalski 79]. That problem is to characterize the events shown in figure 13.

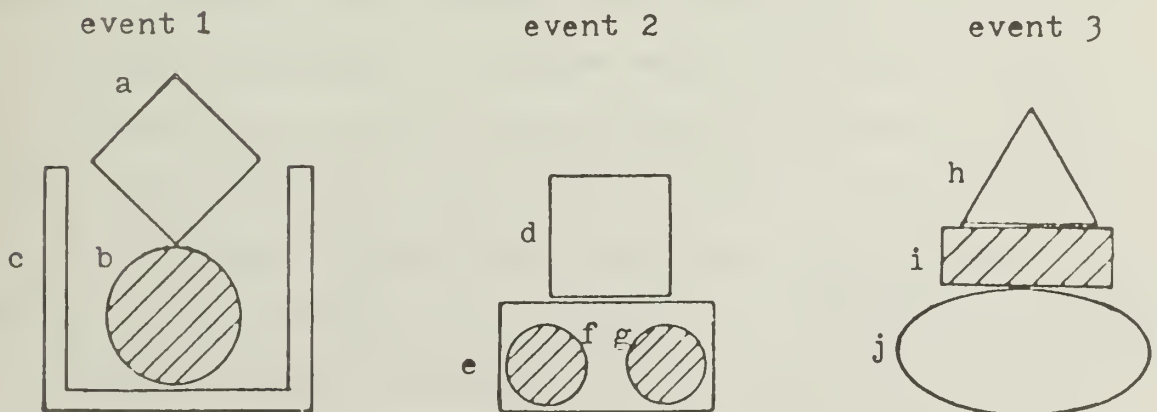
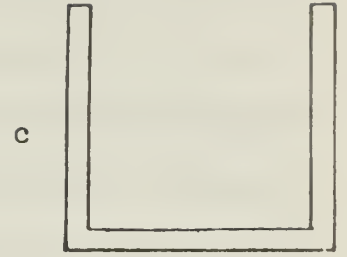
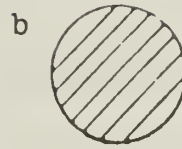
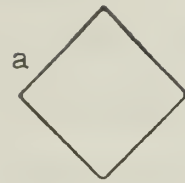


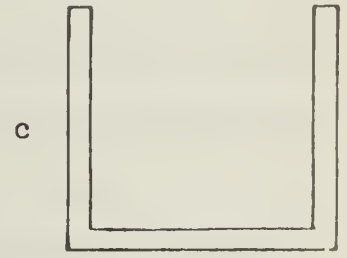
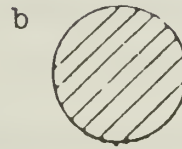
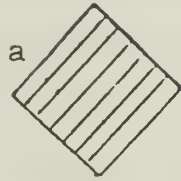
Figure 13

An event is structured when it consists of subevents with similar features (e.g. size, shape, texture) and the relations between subevents (e.g. larger than, ontop, within). To compare events we must first find and compare corresponding subevents, according to their relationships. Consider events 1 and 3 in figure 13. Each of the several objects in each event is a subevent and has observable features of size, shape, and texture. When we compare events 1 and 3 we could compare object a to j, b to h, and c to i, but our natural approach to this task would be to first decide which objects are comparable, and then make the comparisons. Most people would compare objects a and h, b and i, and c and j because a and h are on the top, b and i are in the middle, and c and j are on the bottom. But other lines of reasoning are also valid: compare b and i because they are both shaded, compare a and h because they are both small, compare c and j because they are both large, etc. It is just coincidence that these latter notions of comparability lead to the same mapping of comparable objects. This is rarely true unfortunately and you are directed to the task of comparing events 1 and 2 to realize the difficulty.

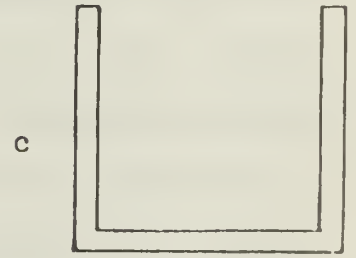
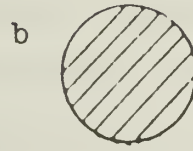
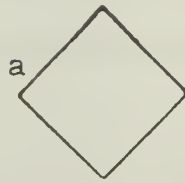
If we ignore the relations between objects in events 1 and 2 the task of comparing them becomes easier. Consider events 1' and 2 in figure 14. We may compare object a to any object d, e, f, or g but it seems most reasonable to compare object a with the object in event 2 which is most



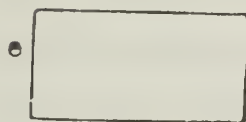
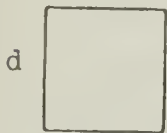
event 1'



event 1''



event 1'''



event 2



Figure 14

similar, i.e. event d. Similarly, event b is best compared to either event f or g, and event c is best compared to event e. If event 1 is given by 1'' in figure 14, the changed texture of object a now makes the comparison of events 1'' and 2 more difficult. Object a is just as much like object d as either f or g and the three choices of a comparable object may have to be carried throughout the induction process. Finally there is the possibility that certain objects may be so dissimilar that it is best to declare them non-comparable. Referring now to the task of comparing events 1''' and 2 of figure 14, if we decided to form pairs of comparable objects (a,d), (b,f), and (c,e) we are left with objects x and g. It would seem foolish to compare x to g just because they are left over after other pairings have been made. Perhaps object x or g represents a unique situation which distinguishes their respective event and which is truly not comparable to any other object. By introducing a similarity threshold limit into the comparable event finding process, objects which are not sufficiently similar can be declared non-comparable. One implementation of non-comparability is the substitution of a special null object for a non-comparable object in the comparable object pairings.

The algorithm below finds comparable objects when presented with n structured events.

Algorithm S:

1. Select an event at random to be the fundamental

event. The algorithm will generate sets of comparable objects, one from each event, for each object in the fundamental event.

2. Let  $m$  be the number of objects in the fundamental event and let  $t$  be the similarity threshold value.

The following steps 3 to 6 are to be repeated  $m$  times to form the  $m$  sets of comparable events, one set for each object in the fundamental event.

3. Compare the fundamental object (i.e. the object of current interest in the fundamental event) with each unclassified object in each event.

4. Find the set of objects of maximum similarity for each event. If the maximum similarity is less than  $t$  then substitute the special null object for the object of maximum similarity.

5. If a single object, or the null object was selected in step 4 it is the comparable object and enters the set of comparable objects being constructed.

6. If several objects were selected in step 4 (i.e. a tie in similarity value) then save the set of alternatives and continue on to process the next fundamental object.

As further object classification proceeds, make the selection firm when only one of the alternatives remains, the others having been assigned to other subsequent object comparability sets.

7. At the end, if any alternative sets remain, make a



firm selection of one object randomly. The algorithm ends with  $m$  sets of  $n$  objects each.

Algorithm S will be illustrated by identifying the comparable objects in the three events shown in figure 13. An english description of the events in figure 13 is:

event 1: "Three objects a, b, and c are arranged with a ontop of b ontop of c. Object a is a medium, clear square. Object b is a medium, shaded circle. Object c is a large, clear Ushape."

event 2: "Four objects d, e, f, and g are arranged with d ontop of e, and f and g within e. Object d is a medium, clear square. Object e is medium, clear rectangle. Objects f and g are small shaded circles."

event 3: "Three objects h, i, and j are arranged with h ontop of i ontop of j. Object h is a medium, clear triangle. Object i is a medium, shaded rectangle. Object j is a large, clear ellipse."

The events will be described formally by the variables size (s-small, m-medium, l-large), texture (c-clear, s-shaded), shape (s-square, c-circle, u-Ushape, r-rectangle, t-triangle, e-ellipse) and on (the value of on is the identity of the object on which it rests). The relation within which applies only to event 2 will not be used. Table 1 gives the formal description of the three events.

event:	1	1	1	2	2	2	2	3	3	3
object:	1	2	3	1	2	3	4	1	2	3
	a	b	c	d	e	f	g	h	i	j
size:	m	m	l	m	m	s	s	m	m	l
texture:	c	s	c	c	c	s	s	c	s	c
shape:	s	c	u	s	r	c	c	t	r	c
on:	2	3	-	2	-	-	-	2	3	-

Table 1



In this application of Algorithm S, the similarity measure will be the count of matching variable values. The value for the variable on is matched in a special way and counts twice. The match score is 1 if the on values are both null (-) or both non-null. An additional point is scored if the value of the on values in the previous object columns match exactly.

We begin to apply algorithm S by selecting the fundamental event. Let it be event 1. Then we proceed to find the set of comparable events for object a. We compare the values of size, texture, shape, on, of object a to those of objects d, e, f, and g (we select d) and to those of objects h, i, and j (we select h). One set of comparable objects is thus {a,d,h}. Next we compare object b to e, f, and g (we select e,f,and g all with a similarity of 2) and we compare object b to objects i and j (we select i). Now object c is compared to objects e,f, and g (we select e) and to object j (we select j). The third set of comparable objects is {c,e,j} and we are left with one alternatives set still containing f and g from which we randomly choose f and the second set of comparable objects becomes {b,f,i}. Each event can now be represented by the values of the variables for an object from each of the three sets of comparable events, using logic system VL<sub>1</sub>. When this data is given to the program AQ7UNI and the results paraphrased in english the description of the events in figure 13 is: "There is a medium sized clear square or triangle ontop of either (a) a small or medium shaded circle or

rectangle or (b) a medium or large clear Ushape, rectangle or ellipse."

Setting the selector threshold low eliminates selectors with multi-valued reference sets and produces the simpler description: "There is a medium sized clear object ontop of a shaded object or a clear object." By interpreting selectors which may not be applicable to all events as possible situations, it may also be said that "The shaded object may be ontop of the clear object."

When algorithm S is applied with event 2 as the fundamental event, a different generalization is formed: "There is a medium sized clear object ontop of another object," or with more detail: "There is a medium sized clear square or triangle ontop of either (a) a medium or large rectangle or Ushape or (b) a circle or ellipse. Object (a) or (b) might be ontop of the other."

When event 3 is the fundamental event, the generalization is: "There is a medium sized clear object ontop of a medium object which may be ontop of another object" and with more detail: "There is a medium sized clear square or triangle ontop of a medium sized circle or rectangle which may be ontop of a small or large ellipse, circle or Ushape." Setting the similarity threshold to 2, the last generalization becomes "There is a medium sized clear square or triangle ontop of a medium sized circle or rectangle which may be ontop of an object which might be

a large clear ellipse or Ushape."

The characterizations of the events in figure 13 produced by algorithm S with AQ7UNI are similar to those of the other methods cited which were studied by Dietterich and Michalski. Some characterizations from their study are:

"There is a medium object ontop of a large, clear object." (Hayes-Roth's method)

"There is a medium object ontop of a large clear object. There is a shaded object and there is a clear object." (Vere's method)

"There is a medium-size circle, rectangle, or square ontop of a large, clear Ushape, rectangle, or ellipse." (Michalski's method)

"There are exactly two clear objects in each event. The top-most object is a medium sized, clear polygon and it is ontop of a large or medium sized circle or rectangle." (Michalski's method with constructive induction)

Many other characterizations are given in Dietterich and Michalski's paper, however the samples given above show the general flavor of the characterizations which can be generated. The last sample above uniquely shows the added power of constructive induction which is a technique not available in the AQ7UNI method.

A summary of the differences of the characterization techniques which have been mentioned is given in figure 15, which appears in [Dietterich & Michalski 79], except for the last column pertaining to the AQ7UNI-with-algorithm-S technique, which appears here for the first time.

Method:	Hayes-Roth	Vere	Buchanan et al.	Michalski	AQ7UNI with S algorithm
Criterion					
Intended Application:	general	general	discovering mass spectrometry rules	general	general
Language:	Parameterized Structured Representation	Quantifier-free FOPL	Chemical model	Variable-valued logic system VL2i	Variable-valued logic system VL1
syntactic concepts:	case frames parameters case labels	literals constants	molecule graph attributes constants in value sets	selectors* descriptors dummy variables constants in value sets	selectors** constants in value sets
operators:	$\wedge$	$\wedge$	$\wedge, V, \text{internal } V$	$\wedge, V, \text{internal } V$	$\wedge, V, \text{internal } V$
Generalization Rules:					
dropping condition?	yes	yes	yes	yes	yes
constants to variables?	yes	yes	yes	yes	no
creating internal $v$ ?	no	no	yes	yes	yes
climbing tree?	no	no	no	yes	yes
closing intervals?	no	no	no	yes	yes
Efficiency:					
comparisons:	22	complete algorithm not known	not applicable	28	16 (in algorithm S)
number of conjunctions generated during search:	20		not applicable	13	not applicable
	6/20=30%		not applicable	4/13=30%	not applicable
Extensibility:					
applications:	speech analysis	none	mass spectrometry	soybean disease diagnosis	general
disjunctive forms?	no	yes	NMR yes	yes	yes
noise immunity:	low	probably good	excellent	very good	thought to be good
problem domain knowledge?	no	limited	yes (Built into program)	yes	limited
constructive induction?	no	limited facility	no	limited facility	no

\*in VL2i, selectors may be simple or complex (involving n-ary functions and predicates) \*\*in VL1 selectors must be simple: [variable = value(s)]

### Comparison of Characterization Methods

(an extension of a table appearing in [Dietterich & Michalski 79])  
Figure 15

### Summary

Inductive program AQ7UNI can characterize any class of events which can be described in the Variable-Valued Logic system  $VL_1$ . The degree of generalization can be controlled by adjustments to the selector threshold and density threshold, and the optimality of the solution can be altered by parameters controlling neighborhood construction and neighborhood judging criteria.

Characterizations of medium degrees of generality usually cause several complexes to be formed, each covering a portion of the events. When disjoint complexes are requested, the complexes describe clusters or subgroups of events in the class which have similar characteristics. Unique events tend to fall into the smallest subgroups because they are the most difficult to describe generally.

The great flexibility of the program with several control parameters and the wide range of characterization problems and solution requirements makes experimentation the only technique for exploring the range of possible characterizations in order to find those which are useful.

AQ7UNI has no facilities for constructive induction nor can it handle problems involving structured events. Sometimes these two limitations can be overcome by manually introducing new variables (in lieu of constructive induction) or transforming a structured-event problem into a  $VL_1$ -expressable one (e.g. via algorithm S).



## REFERENCES

- Buchanan, B. G., Feigenbaum, E. A., "Dendral and Meta-Dendral, Their Applications Dimension," *Artificial Intelligence*, Vol 11, pp5-24, 1978.
- Dietterich, T. G., Michalski, R. S., "Learning and Generalization of Characteristic Descriptions: Evaluation Criteria and Comparative Review of Selected Methods," submitted for publication to the Sixth International Joint Conference on Artificial Intelligence, August, 1979.
- Hayes-Roth, Frederick, "Patterns of Induction and Associated Knowledge Acquisition Algorithms," Department of Computer Science, Carnegie-Mellon University, May 1976.
- Larson, J., Michalski, R. S., "AQVAL/1(AQ7) User's Guide and Program Description," Department of Computer Science report number 731, University of Illinois, Urbana Illinois, June 1975.
- Larson, James G., "Inductive Inference in the Variable Valued Predicate Logic System VL21: Methodology and Computer Implementation," Department of Computer Science report number 869, University of Illinois, Urbana Illinois, May 1977.
- Michalski, R. S., "AQVAL/1--Computer Implementation of a Variable-Valued Logic System and the Application to Pattern Recognition," Proceedings of the First International Joint Conference on Pattern Recognition, Washington, D. C., October 30-November 1, 1973.
- Michalski, R. S., "VARIABLE-VALUED LOGIC: System VL<sub>1</sub>," 1974 International Symposium on Multiple-Valued Logic, West Virginia University, Morgantown, West Virginia, May 29-31, 1974.
- Michalski, R. S., "Variable-Valued Logic and its Application to Pattern Recognition and Machine Learning," chapter in the monograph: Multiple-Valued Logic and Computer Science, ed. David Rine, North-Holland publishers, 1975.

- Michalski, R. S., "Toward Computer-Aided Induction: A Brief Review of Currently Implemented AQVAL Programs," Department of Computer Science report number 874, University of Illinois, Urbana Illinois, May 1977(a).
- Michalski, R. S., "A SYSTEM OF PROGRAMS FOR COMPUTER-AIDED INDUCTION: A SUMMARY," 5th International Joint Conference on Artificial Intelligence, MIT, Boston, Massachusetts, August 1977(b).
- Michalski, R. S., "A Planar Geometrical Model for Representing Multidimensional Discrete Spaces and Multiple-valued Logic Functions," Department of Computer Science report number 897, University of Illinois, Urbana Illinois, January 1978.
- Michalski, R. S., and Larson, J. B., "SELECTION OF MOST REPRESENTATIVE TRAINING EXAMPLES AND INCREMENTAL GENERATION OF  $VL_1$  HYPOTHESES: the underlying methodology and the description of programs ESEL and AQ11," Department of Computer Science report Number 867, University of Illinois, Urbana Illinois, May 1978.
- Michalski, R. S., "STUDIES IN COMPUTER INDUCTION AND PLAUSIBLE INFERENCE," a research proposal submitted to the National Science Foundation, Intelligent Systems Program, Computer Science Section, Division of Mathematical and Computer Sciences (1979).
- Stepp, Robert, "The Uniclass Inductive Program AQ7UNI: Program Implementation and User's Guide," Department of Computer Science report number 949, University of Illinois, Urbana Illinois, April 1979.
- Vere, S. A., "Multilevel Counterfactuals for Generalizations of Relational Concepts and Productions," Department of Information Engineering, University of Illinois, Chicago Circle, 1978.
- Winston, P. H., "Learning Structural Descriptions from Examples," Technical Report AI TR-231, MIT AI Lab, Cambridge, Massachusetts, 1970.





<b>BIBLIOGRAPHIC DATA SHEET</b>	1. Report No. UIUCDCS-R-79-982	2.	3. Recipient's Accession No.
4. Title and Subtitle Learning Without Negative Examples via Variable-Valued Logic Characterizations: The Uniclass Inductive Program AQ7UNI		5. Report Date July 1979	
7. Author(s) Robert Stepp		6.	
9. Performing Organization Name and Address Department of Computer Science University of Illinois Urbana, IL 61801		8. Performing Organization Rept. No.	
12. Sponsoring Organization Name and Address National Science Foundation		10. Project/Task/Work Unit No.	
		11. Contract/Grant No. NSF MCS 79-06614	
		13. Type of Report & Period Covered	
		14.	
15. Supplementary Notes			
16. Abstracts This paper describes the underlying theory, internal logic, and evaluation of an inductive program AQ7UNI which accepts a set of symbolic descriptions (events) of arbitrary objects and produces a general description (characterization) of the set. The events are attributevalue lists and the resulting characterizations are expressed in a simple yet powerful formal language VL <sub>1</sub> (Variable-valued Logic system 1 [Michalski 74, 75]), which is a form of monadic predicate calculus.			
17. Key Words and Document Analysis. 17a. Descriptors  Computer Induction Machine Learning without Teacher Variable-valued Logic Characteristic descriptions			
17b. Identifiers/Open-Ended Terms			
17c. COSATI Field/Group			
18. Availability Statement Unlimited		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 57
		20. Security Class (This Page) UNCLASSIFIED	22. Price











FEB 20 1981



UNIVERSITY OF ILLINOIS-URBANA



3 0112 045361299